Interrelated Dense Subgraph Detection in Multilayered Networks

Wenjie Feng^{1,2,†,*}, Li Wang^{2,†}, Bryan Hooi^{1}, See Kiong Ng^{1}, Shenghua Liu^{2,*}

^{1}*Institute of Data Science, National University of Singapore, Singapore*
^{2}*Institute of Computing Technology, Chinese Academic of Sciences, China*
{wenchiehfeng.us, wl.tsinghua.ee}@gmail.com, {dcsbhk, seekiong}@nus.edu.sg, liushenghua@ict.ac.cn

***Abstract*—We provide detailed information about related algorithms and real-world datasets, give more experimental results and analysis, and more case studies of real-world networks, to support the main paper.**

Appendix

A. Densest Subgraph Detection

Here, we provide details of some dense subgraph detection algorithms, which consider the undirected graph $G = (\mathcal{V}, \mathcal{E})$.

Algorithm 1 summarizes the greedy detection algorithm GreedySolver. It takes the original nodeset $\mathcal{V}$ as the starting point, then greedily removes the node with the smallest degree from the graph G (i.e., the selection step in Line 3), and returns the densest subgraph among the shrinking sequence of subgraphs created by the procedure. When using the *priority tree* data structure to manage the nodes during the peeling process, the time complexity of the GreedySolver algorithm is $O(|\mathcal{E}| \log |\mathcal{V}|)$.

Algorithm 1: GreedySolver: Densest Subgraph Detection with Greedy algorithm

```
Input: Undirected graph $G = (\mathcal{V}, \mathcal{E})$; density metric $\rho(\cdot)$
Output: Nodeset of the optimal densest subgraph of $G$.
1 $\mathcal{V}^* \leftarrow \mathcal{V}$
2 while $\mathcal{V} \neq \emptyset$ do
      ▷ find the vertex to maximize the density
3     $u^* \leftarrow \arg\max_u \rho_G(\mathcal{V} \setminus \{u\})$
      ▷ $\mathcal{V} \setminus \{u^*\}$ is the remaining node-set without $u$
4     Remove $u^*$ and all its adjacent edges from $G$
5     if $\rho_G(\mathcal{V}) > \rho_G(\mathcal{V}^*)$ then $\mathcal{V}^* \leftarrow \mathcal{V}$ ;
6 return $\mathcal{V}^*$.
```

Algorithm 2 summarizes how to detect dense subgraph with a MF-Solver over the graph G, and MF-Solver is the decomposition-based approximate detection method, e.g., EigenSolver and NMFSolver. MF-DSD selects the top-ranked vertices based on the scores of each factor $\mathbf{U}(:, k)$ with a given threshold Δ (Line 5), and returns the subgraph with the highest density derived from those vertices. A commonly used threshold is $\Delta = \frac{1}{|\mathcal{V}|}$, which is from the random graph assumption, i.e., the link between two nodes exists with equal probability. The total time complexity is $O(K|\mathcal{V}|)+O(\text{MF-Solver})$, which is dominated by the second term.

Algorithm 2: MF-DSD: Matrix Factorization based Dense Subgraph Detection over an aggregated-graph

```
Input: Undirected graph $G = (\mathcal{V}, \mathcal{E})$; density metric
       $\rho(\cdot)$; MF-Solver; factorization rank $K$; vector
       truncation threshold $\Delta$.
Output: Nodeset of the optimal densest subgraph of $G$.
1 $\mathcal{V}^* \leftarrow \emptyset$
  ▷ get top-$K$ factors $\mathbf{U}$ with the MF-Solver.
2 $\mathbf{U} \leftarrow$ MF-Solver$(\mathbf{A}_G, K)$
3 for $k \leftarrow 1$ to $K$ do
4     $\mathbf{U}(:, k) = \frac{\mathbf{U}(:,k)}{||\mathbf{U}(:,k)||_2}$        ▷ vector normalization.
5     $\bar{\mathcal{V}} = \{v \,|\, |\mathbf{U}(v, k)| \geq \Delta, \forall\, v \in [|\mathcal{V}|]\}$
6     if $\rho_G(\bar{\mathcal{V}}) > \rho_G(\mathcal{V}^*)$ then
7         $\mathcal{V}^* \leftarrow \bar{\mathcal{V}}$
8 return $\mathcal{V}^*$.
```

B. Density Metric Analysis

We propose the *joint density for the multilayered networks* in Definition 1 as a flexible density metric, where the hyperparameters $\{\gamma_{i,j}\}$s control the contribution of the cross-layer $\mathbf{C}_{i,j}$ to the total density and different cross-layers can also use distinct γs (we only adopt the same value for $\gamma_{i,j}$ in our experiments). When $\gamma = 0$, the cross-layer links are not considered, and the detected result might be (near-) free dense subgraph (no guarantee for the interrelation), at this point, it is equivalent to applying the DSD algorithms (i.e., DSDSolver) layer by layer alone; and when $\gamma \to \infty$, the contribution of within-layer links becomes very weak, the detected dense subgraphs are mainly composed of the cross-layer dependencies (almost equivalent to $g + 1$-partite graph); given a proper value of $0 < \gamma \ll \infty$, there is a trade-off of the contribution of within-layers and cross-layers for detecting the interrelated dense subgraph.

C. Proof of Lemma 1.

We analyze the objective function w.r.t. $\mathbf{U}_i$ separately.

$$L(\mathbf{U}_i) = \sum_{j=1}^{g} \beta_{i,j} \left\| \mathbf{C}_{i,j} - \mathbf{U}_i \mathbf{\Sigma}_{i,j} \mathbf{U}_j' \right\|_F^2 + \left\| \mathbf{A}_i - \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i' \right\|_F^2$$

Following the standard theory of constrained optimization, we introduce the Lagrangian multipliers $\mathbf{\Theta}$ (a symmetric matrix of size $n_i \times n_i$), then the regularized objective function becomes

$$\begin{aligned} L(\mathbf{U}_i) =& \sum_{j=1}^{g} \beta_{i,j} \left\| \mathbf{C}_{i,j} - \mathbf{U}_i \mathbf{\Sigma}_{i,j} \mathbf{U}_j' \right\|_F^2 + \left\| \mathbf{A}_i - \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i' \right\|_F^2 \\ &+ \operatorname{tr} \left[\mathbf{\Theta} (\mathbf{U}_i' \mathbf{U}_i - \mathbf{I}) \right] \end{aligned}$$

We can get the gradient with respect to $\mathbf{U}_i$ as

$$\begin{aligned} \frac{\partial L(\mathbf{U}_i, \mathbf{\Theta})}{\partial \mathbf{U}_i} =& \sum_{j=1}^{g} 2\beta_{i,j} \left(\mathbf{U}_i \mathbf{\Sigma}_{i,j} \mathbf{U}_j' \mathbf{U}_j \mathbf{\Sigma}_{i,j} - \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} \right) \\ &+ 4 \left(\mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i' \mathbf{U}_i \mathbf{\Lambda}_i - \mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i \right) + 2\mathbf{U}_i \mathbf{\Theta}. \end{aligned}$$

Using the KKT complementary condition for the non-negativity of $\mathbf{U}_i$, it gives

$$\frac{\partial L \left(\mathbf{U}_i, \mathbf{\Theta} \right)}{\partial \mathbf{U}_i} \otimes \mathbf{U}_i = 0.$$

Thus, we give the update rule for $\mathbf{U}_i$ as

$$\mathbf{U}_i \leftarrow \mathbf{U}_i \otimes \frac{\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i}{\mathbf{U}_i \left(\sum_{j=1}^{g} \beta_{i,j} \mathbf{\Sigma}_{i,j} \mathbf{U}_j' \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{\Lambda}_i \mathbf{U}_i' \mathbf{U}_i \mathbf{\Lambda}_i + \mathbf{\Theta} \right)}$$

According to [Ding, Li, Peng, and ParkDing et al.2006], we set the Lagrangian multipliers $\mathbf{\Theta}$ as

$$\begin{aligned} \mathbf{\Theta} =& \mathbf{U}_i' \left(\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i \right) \\ &- \left(\sum_{j=1}^{g} \beta_{i,j} \mathbf{\Sigma}_{i,j} \mathbf{U}_j' \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{\Lambda}_i \mathbf{U}_i' \mathbf{U}_i \mathbf{\Lambda}_i \right), \end{aligned}$$

Combining the scaling exponent as suggested in [Welling and WeberWelling and Weber2001], which is set to be $\frac{1}{2}$ here, and keeping other matrices fixed, we derive the multiplicative update rule of $\mathbf{U}_i$ as follows,

$$\mathbf{U}_i \leftarrow \mathbf{U}_i \otimes \frac{\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i}{\mathbf{U}_i \mathbf{U}_i' \left(\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i \right)}.$$

As [Welling and WeberWelling and Weber2001] note the need for a scaling exponent, we adopt $\frac{1}{2}$ to the above result as our update rule,

$$\mathbf{U}_i \leftarrow \mathbf{U}_i \otimes \left(\frac{\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i}{\mathbf{U}_i \mathbf{U}_i' \left(\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i \right)} \right)^{\frac{1}{2}}.$$

Similarly, for the diagonal matrices $\mathbf{\Lambda}_i$ and $\mathbf{\Sigma}_{i,j}$, we have

$$\begin{aligned} \mathbf{\Lambda}_i &\leftarrow \mathbf{\Lambda}_i \otimes \left(\frac{\mathbf{U}_i' \mathbf{A}_i \mathbf{U}_i}{\mathbf{U}_i' \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i' \mathbf{U}_i} \right)^{\frac{1}{2}}, \\ \mathbf{\Sigma}_{i,j} &\leftarrow \mathbf{\Sigma}_{i,j} \otimes \left(\frac{\mathbf{U}_i' \mathbf{C}_{i,j} \mathbf{U}_j}{\mathbf{U}_i' \mathbf{U}_i \mathbf{\Sigma}_{i,j} \mathbf{U}_j' \mathbf{U}_j} \right)^{\frac{1}{2}}. \end{aligned}$$

Moreover, the convergence of the CoNF algorithm is guaranteed by Theorem 5 in [Ding, Li, Peng, and ParkDing et al.2006], refer to the original paper for detailed proof.

D. Dataset Information.

We list the description of the real-world multilayered networks as follows, and their statistical information is summarized in Tab. 2.

- *AMINER* [Tang, Zhang, Yao, Li, Zhang, and SuTang et al.2008]: a three-layered network in the academic collaboration domain, which contains co-authorship, paper-citation, and a venue-citation within networks. The cross-layer dependencies are naturally generated from who-writes-which paper and which venue-publishes-which paper [Chen, He, Bliss, and TongChen et al.2017].
- *BIO* [Razick, Magklaras, and DonaldsonRazick et al.2008]: a three-layer CTD (Comparative Toxicogenomics Database) network in the biological domain, contains chemical-similarity, gene-similarity, and disease-similarity as within networks; which chemical-influences-which gene, which chemical-cures-which disease, and which gene-related-which disease constitute the cross dependencies of different layers.
- *DBLP* [Tang, Zhang, Yao, Li, Zhang, and SuTang et al.2008]: a three-layered dataset similar to *AMINER*, which has different sizes and is collected for different time spans.
- *INFRA-3* [Chen, Tong, Xie, Ying, and HeChen et al.2016]: a three-layered network in the critical infrastructure domain, which contains an airport network, an autonomous system network, and a power grid. The cross-layer dependencies between different layers are generated from geographic proximity.
- *INFRA-5* [Chen, Tong, Xie, Ying, and HeChen et al.2016]: a five-layered network consists of one internet network and four regional power grids. The regional power grids are partitioned by macro-regions, and the dependencies between these layers are based on the power transfer lines. See detailed information in [Chen, Tong, Xie, Ying, and HeChen et al.2016].

Note that *INFRA-5* is only for the comparison with DESTINE; the (# of nodes, # of within-layer links, # of cross-layer links) of INFA-5 corresponds to (249, 379, 565) respectively. Due to the small size, it was not used in our main experiments.

E. Detection performance under camouflage.

As in Sec. 5.5, we apply all detection algorithms to the IDSD problem over different networks under camouflage.

The configuration of *SF network* keeps the same as Sec. 5.4.1; we also follow the same way as for ER network in Sec. 5.3 to generate the camouflage, where the link probability is also p_{cam}. Here, we report the *F score* of the detection results for each layer as well, and merge those layers if they have the same results.

Table I and Table II show the detection performance with the change of injection probability p_{gt} of the injected interrelated dense subgraphs under $p_{cam} = 1.0$.

For the *sub-BIO network*, INDUEN with $\gamma = 15$ nearly achieves the best result with $F \approx 1.0$ while in some cases

TABLE I
F scores on *sub-BIO* network with different injection density under camouflage ($p_{cam} = 1.00$). $\gamma = 15$ for InDuen.

Method	k-th layer	Injection density p_{gt}					
		0.10	0.20	0.30	0.40	0.50	0.60
Destine	1,2,3	0.00	0.00	0.00	0.00	0.67	0.67
InDuen / Exp.	1	**1.00**	**1.00**	0.89	**1.00**	0.91	0.86
	2	0.99	**1.00**	**1.00**	**1.00**	0.84	0.76
	3	**1.00**	**1.00**	**1.00**	**1.00**	0.84	0.76
InDuen	1	0.98	0.98	**0.98**	0.98	**0.98**	**0.98**
	2	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	3	0.98	0.98	0.99	0.99	**0.99**	**0.99**

* Here, OQC,OQC^a, Fraudar, Fraudar^a, MF-DSD,MF-DSD^a and Greedy are ignored since their detection results are all 0.00.

for the 1st and 3rd layer, Expander introduces some non-target node, which slightly damages its precision and F score, compared with InDuen/Exp.; but these two algorithms still outperform all others. Many baseline methods fail to detect any target, which is similar to the results over ER network; Destine can detect some results when the injection density is not less than 0.50, where it cannot distinguish camouflage and the target and gets $F = 0.67$ (precision = 0.50, recall = 1.0).

For the *SF network*, InDuen with $\gamma = 20$ achieves $F = 1.0$ and outperforms all baselines, and the Expander greatly improves its result, compared with InDuen/Exp.; while the latter is also better than or equal to other baselines; In addition, the underlying network model also influences the performance of InDuen/Exp. compared with Tab. 5 in main paper, especially for low injection densities. Fraudar and Greedy cannot detect any targets until $p_{gt} = 0.60$; OQC detects few targets for all layers, except for the 3rd layer when $p_{gt} = 0.60$; MF-DSD can find some targets for all layers, even for low injection density. For the aggregated graph, OQC^a and Fraudar^a do not find any target, MF-DSD^a can only detect results for the 3rd layer with a larger size. For all layers, the performance of Destine is insensitive to the injection density here, which is consistent with Fig. 3(b); and it has better performance for smaller-size graphs.

a) Influence of camouflage strength:: In addition, to show the influence of the strength of camouflage (i.e., p_{cam}) on the performance and serve as a comparison, we use the *ER network* and change p_{cam} to be 0.5 while keeping other setting the same as Sec. **??**. Then, we show the detection results in Tab. 5.3.1 As we can see, InDuen still achieves perfect detection performance and outperforms all baselines significantly. When $p_{cam} = p_{gt}$, InDuen/Exp. has the worst results since that CoNF can not well distinguish the camouflage and target if their density is very similar, and thus the improvement of neighbor booster (Expander) for InDuen is more prominent; in addition, InDuen/Exp. drops a lot for low injection density here is also because of distinguishing between p_{cam} and p_{gt}, compared with Tab.5. With the weaken camouflage, other baselines have a little better performance than the case of $p_{cam} = 1.0$, but most of them still only find very few targets when $p_{gt} \leq 0.40$; MF-DSD^a and Destine also have a lower detection density, OQC^a and Fraudar cannot find any target until $p_{gt} \geq 0.50$.

TABLE II
F scores on *SF network* with different injection density under camouflage ($p_{cam} = 1.00$). $\gamma = 20$ for InDuen.

Method	k-th layer	Injection density p_{gt}					
		0.10	0.20	0.30	0.40	0.50	0.60
OQC	1	0.02	0.02	0.02	0.02	0.04	0.04
	2	0.05	0.06	0.06	0.07	0.08	0.09
	3	0.05	0.06	0.06	0.06	0.09	0.67
OQC^a	1,2,3	0.00	0.00	0.00	0.00	0.00	0.00
Fraudar	1	0.00	0.00	0.00	0.00	0.00	0.00
	2,3	0.00	0.00	0.00	0.00	0.00	0.67
Fraudar^a	1,2,3	0.00	0.00	0.00	0.00	0.00	0.00
MF-DSD	1	0.13	0.15	0.23	0.31	0.67	0.67
	2	0.19	0.23	0.29	0.42	0.67	0.67
	3	0.21	0.28	0.35	0.52	0.66	0.67
MF-DSD^a	1,2	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.58	0.64	0.66	0.67	0.67	0.67
Greedy	1,2	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	0.00	0.67
Destine	1	0.65	0.65	0.65	0.65	0.65	0.65
	2	0.60	0.60	0.60	0.60	0.60	0.60
	3	0.55	0.55	0.55	0.55	0.55	0.55
InDuen / Exp.	1	0.64	0.66	0.67	0.64	0.66	0.67
	2	0.67	0.66	0.68	0.68	0.67	0.67
	3	0.68	0.67	0.69	0.69	0.71	0.70
InDuen	1,2,3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

TABLE III
F scores on *ER network* with different injection density under camouflage ($p_{cam} = 0.50$). $\gamma = 15$ for InDuen.

Method	k-th layer	Injection density p_{gt}					
		0.10	0.20	0.30	0.40	0.50	0.60
OQC	1,2,3	0.18	0.18	0.18	0.18	0.18	0.18
OQC^a	1,2,3	0.00	0.00	0.00	0.00	0.00	**1.00**
Fraudar	1,2	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	0.00	**1.00**
Fraudar^a	1,2,3	0.18	0.18	0.18	0.18	0.18	0.18
MF-DSD	1,2	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.18	0.18	0.18	0.18	0.18	0.18
MF-DSD^a	1	0.00	0.09	0.67	0.67	0.67	0.67
	2	0.00	0.07	0.67	0.67	0.67	0.67
	3	0.00	0.05	0.67	0.67	0.67	0.67
Greedy	1	0.00	0.00	0.00	0.00	**1.00**	0.66
	2	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.21	0.23	0.23	0.26	0.67	0.18
Destine	1,2,3	0.00	0.00	0.67	0.67	0.67	0.67
InDuen/ Exp	1	0.62	0.59	0.57	0.58	0.49	0.56
	2	0.56	0.55	0.58	0.54	0.66	0.56
	3	0.61	0.59	0.57	0.65	0.52	0.58
InDuen	1,2,3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

F. Case study: Drug-Disease Networks

We chose another two-layer network from gu2022redda for the 'drug-disease', which includes 894 drugs, 454 diseases, and 2704 drug-disease associations. In the within-layer similarity graphs, we keep an edge if the similarity between two drugs/diseases is not less than 0.30.

Figure 1 shows the detected interrelated dense sub-network of InDuen with $\gamma = 1.2$. Fig. 1(a) is the global view of the interrelated dense subgraph, Fig. 1(b)-1(c) illustrate the within-layer graph for drugs' and diseases' similarity networks with node labels. We can get the name and detailed information from the databases[1] about the drugs and diseases based on the DrugBank ID and Mesh ID. Tab. IV-V list the basic information about the drugs and diseases contained in Fig. 1. The dense-connected components like this will benefit related fields, like disease diagnosis, drug discovery, etc.

G. Comparison with DESTINE.

For a fair comparison with DESTINE, we construct the same experiment settings and report the detection results of different methods.

1) Detection of Injected Interrelated Cliques: We test the performance for detecting the injected interrelated cliques on synthetic networks (*ER* and *SF*). The injection strategy here is slightly different from the description in our main text, and see Xu2021DESTINEDS for the detail settings. Table VI shows the detection results of different methods, and p or m of the underlying model for each layer is provided.

As we can see, INDUEN with $\gamma = 30$ consistently achieves the best results with $F = 1.0$ for each layer in both networks and outperforms all baselines. Therefore, our method can adapt to networks with different underlying models.

For the *ER networks*, DESTINE, MF-DSD, and NMF also have the best performance for easier detection cases, e.g., $p = 0.1$ and 0.2, while becoming inferior for the within layer with more links (e.g., $p = 0.6$), where NMF even closes to the worst; OQC and FRAUDAR get worse results. For the *SF networks*, the conclusion is similar, FRAUDAR, DESTINE and MF-DSD perform as well as INDUEN; NMF just handles those easier cases. In addition, we also use the aggregated graph as the input for OQC, FRAUDAR, and MF-DSD, they also underperformance for most of this setting, where MF-DSD even misses hitting any target.

Besides, we also test the performance of INDUEN with different DSDSOLVERs on these synthetic networks, that is, replacing GREEDYSOLVER with MAXFLOWSOLVER and MAXCLIQUESOLVER, it turns out that they all archieve the same performance. INDUEN using GREEDYSOLVER with Layer-By-Layer detection strategy also gets the results of $F = 1.0$.

2) Detection for Real Networks: Here, we use INDUEN with the Layer-By-Layer strategy (also means $\gamma = 0$) and apply to detect dense subgraphs in each layer over the same networks used by DESTINE, i.e., AMINER, BIO, INFRA-3, and INFRA-5. Table VII shows the detection results, including the edge ratio density and size of the dense subgraphs of each detection method.

We can see that INDUEN can always detect the subgraph with the highest density. OQC tends to detect subgraphs with large sizes and small densities. DESTINE can find near-clique subgraphs but may ignore some important nodes at the *Author* and *Gene* layers compared to INDUEN.

[1] Drug information from https://www.drugs.com/, and disease information from U.S. National Library of Medicine https://id.nlm.nih.gov/mesh/.

REFERENCES

[Chen, He, Bliss, and TongChen et al.2017] Chen Chen, Jingrui He, Nadya Bliss, and Hanghang Tong. 2017. Towards optimal connectivity on multi-layered networks. *IEEE transactions on knowledge and data engineering* 29, 10 (2017), 2332–2346.

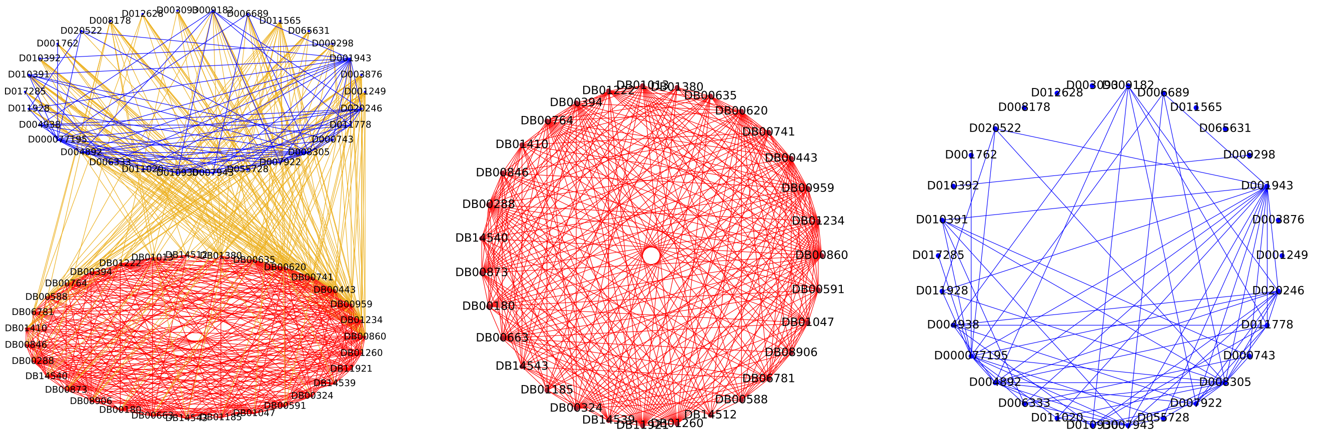
[Chen, Tong, Xie, Ying, and HeChen et al.2016] C. Chen, Hanghang Tong, Lei Xie, Lei Ying, and Qing He. 2016. FASCINATE: Fast Cross-Layer Dependency Inference on Multi-layered Networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[Ding, Li, Peng, and ParkDing et al.2006] Chris Ding, Tao Li, Wei Peng, and Haesun Park. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 126–135.

[Razick, Magklaras, and DonaldsonRazick et al.2008] Sabry Razick, George Magklaras, and Ian M Donaldson. 2008. iRefIndex: a consolidated protein interaction database with provenance. *BMC bioinformatics* 9, 1 (2008), 1–19.

[Tang, Zhang, Yao, Li, Zhang, and SuTang et al.2008] Jie Tang, Jing Zhang, Limin Yao, Juan-Zi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: extraction and mining of academic social networks. In *KDD*.

[Welling and WeberWelling and Weber2001] Max Welling and Markus Weber. 2001. Positive tensor factorization. *Pattern Recognition Letters* 22, 12 (2001), 1255–1261.



(a) Interrelated dense subgraph extracted by InDuen.

(b) Drugs' similarity within-layer graph in (a).

(c) Diseases' similarity within-layer graph in (a).

Fig. 1. Interrelated dense subgraph patterns in the two-layer Drug-Disease networks. (a) shows the detection result of InDuen with $\gamma = 1.2$, the edge density of the drug-disease association network is 0.19. (b) and (c) give the within-layer graph view with node labels of the drugs' similarity and diseases' similarity network of (a) with edge density 0.62 and 0.17, respectively. Here, 'DBXXXX' is the Drugbank ID of some drug, e.g., DB00635 denotes Prednisone with the chemical formula as $C_{21}H_{26}O_5$, and 'D0XXXXX' is the Mesh ID of some disease, e.g., D065631 for Rhinitis, Allergic.

TABLE IV
Drugs Information for the Drug-Disease Network.

DrugBank ID	Name	Chemical Formula	DrugBank ID	Name	Chemical Formula
DB01260	Desonide	C24H32O6	DB00443	Betamethasone	C22H29FO5
DB00288	Amcinonide	C28H35FO7	DB00741	Hydrocortisone	C21H30O5
DB01222	Budesonide	C25H34O6	DB00324	Fluorometholone	C22H29FO4
DB00764	Mometasone	C22H28Cl2O4	DB00846	Flurandrenolide	C24H33FO6
DB00635	Prednisone	C21H26O5	DB01185	Fluoxymesterone	C20H29FO3
DB01410	Ciclesonide	C32H44O7	DB01380	Cortisone acetate	C23H30O6
DB11921	Deflazacort	C25H31NO6	DB14512	Mometasone furoate	C27H30Cl2O6
DB00180	Flunisolide	C24H31FO6	DB00959	Methylprednisolone	C22H30O5
DB00873	Loteprednol	C21H27ClO5	DB08906	Fluticasone furoate	C27H29F3O6S
DB00663	Flumethasone	C22H28F2O5	DB01013	Clobetasol propionate	C25H32ClFO5
DB01047	Fluocinonide	C26H32F2O7	DB00591	Fluocinolone acetonide	C24H30F2O6
DB00860	Prednisolone	C21H28O5	DB00588	Fluticasone propionate	C25H31F3O5S
DB06781	Difluprednate	C27H34F2O7	DB14539	Hydrocortisone acetate	C23H32O6
DB00620	Triamcinolone	C21H27FO6	DB14540	Hydrocortisone butyrate	C25H36O6
DB01234	Dexamethasone	C22H29FO5	DB14543	Hydrocortisone probutate	C28H40O7
			DB00394	Beclomethasone dipropionate	C28H37ClO7

TABLE V
Diseases Information for the Drug-Disease Network.

Mesh ID	Name	Mesh ID	Name	Mesh ID	Name
D001249	Asthma	D011778	Q Fever	D006333	Heart Failure
D011565	Psoriasis	D010930	Plague	D000743	Anemia, Hemolytic
D009298	Nasal Polyps	D001762	Blepharitis	D020522	Lymphoma, Mantle-Cell
D001943	Breast Neoplasms	D010392	Pemphigus	D007943	Leukemia, Hairy Cell
D065631	Rhinitis, Allergic	D007922	Leptospirosis	D055728	Primary Myelofibrosis
D003876	Dermatitis, Atopic	D017285	Polymyositis	D010391	Pemphigoid, Bullous
D006689	Hodgkin Disease	D011928	Raynaud Disease	D008305	Malignant Hyperthermia
D009182	Mycosis Fungoides	D020246	Venous Thrombosis	D011020	Pneumonia, Pneumocystis
D003093	Colitis, Ulcerative	D004938	Esophageal Neoplasms	D008178	Lupus Erythematosus, Cutaneous
D012628	Dermatitis, Seborrheic	D004892	Erythema Multiforme	D000077195	Squamous Cell Carcinoma of Head and Neck

TABLE VI
F scores for detecting the injected cliques on the synthetic data.

Data	k**-th layer**	OQC	OQC^a	Fraudar	Fraudar^a	MF-DSD	MF-DSD^a	NMF	Destine	InDuen
ER	$p = 0.1$	0.54	0.00	0.18	0.00	**1.00**	0.00	**1.00**	**1.00**	**1.00**
	$p = 0.2$	0.03	0.00	0.18	0.00	**1.00**	0.00	**1.00**	**1.00**	**1.00**
	$p = 0.6$	0.03	0.18	0.18	0.18	0.41	0.00	0.05	0.85	**1.00**
SF	$m = 20$	0.41	0.00	**1.00**	0.00	**1.00**	0.00	**1.00**	**1.00**	**1.00**
	$m = 40$	0.29	0.00	**1.00**	0.00	**1.00**	0.00	0.90	**1.00**	**1.00**
	$m = 60$	0.25	**1.00**	**1.00**	**1.00**	**1.00**	0.00	0.85	**1.00**	**1.00**

* MF-DSD utilizes SVDSolver; ‘^{a}’ labeled methods mean the detection results for the aggregated graph.

TABLE VII
Evaluate over the real-world networks. InDuen with Layer-By-Layer strategy.

Dataset	**Layer**	**Edge Density**					**Size**				
		OQC	NMF	SVD	Destine	InDuen	OQC	NMF	SVD	Destine	InDuen
AMINER	Paper	0.71	0.61	0.01	0.90	**1.00**	10	8	504	5	5
	Venue	0.53	0.94	0.09	0.94	**1.00**	48	20	172	18	13
	Author	**1.00**	**1.00**	0.00	**1.00**	**1.00**	28	27	3,539	26	28
BIO	Chem	**1.00**	**1.00**	0.19	**1.00**	**1.00**	91	91	318	91	91
	Gene	0.43	0.29	0.01	0.97	**1.00**	73	55	3,469	9	14
	DZ	0.70	0.89	0.08	**1.00**	**1.00**	119	89	460	59	58
INFRA-3	AP	0.60	0.92	0.05	0.99	**1.00**	82	44	377	34	31
	AS	0.51	0.76	0.03	**1.00**	**1.00**	13	7	151	4	4
	Power	0.55	0.80	0.01	0.97	**1.00**	51	23	1,045	15	13
INFRA-5	P1	0.53	**1.00**	0.13	**1.00**	**1.00**	6	3	18	3	3
	P2	0.47	0.18	0.06	0.67	**1.00**	6	11	36	3	3
	P3	0.50	**1.00**	0.13	**1.00**	**1.00**	5	3	13	3	3
	P4	0.67	0.20	0.09	**1.00**	**1.00**	4	10	22	3	3
	Net	0.50	0.25	0.12	0.67	**1.00**	4	8	15	4	3