FG$_\ell$T: Fast Graphlet Transform

Dimitris Floros[†], Nikos Pitsianis[†‡], and Xiaobai Sun[‡]

[†]Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
[‡]Department of Computer Science, Duke University, Durham, NC 27708, USA

August 30, 2020

1 Summary

We provide FG$_\ell$T, a `C/C++` multi-threading library, for Fast Graphlet Transform of large, sparse, undirected networks/graphs. The graphlets in dictionary Σ_{16}, shown in Figure 1, are used as encoding elements to capture topological connectivity quantitatively and transform a graph $G = (V, E)$ into a $|V| \times 16$ array of graphlet frequencies at all vertices. The 16-element vector at each vertex represents the frequencies of induced subgraphs, incident at the vertex, of the graphlet patterns. The transformed data array serves multiple types of network analysis: statistical or/and topological measures, comparison, classification, modeling, feature embedding and dynamic variation, among others. The library FG$_\ell$T is distinguished in the following key aspects. (1) It is based on the fast, sparse and exact transform formulas in [6], which are of the lowest time and space complexities among known algorithms, and, at

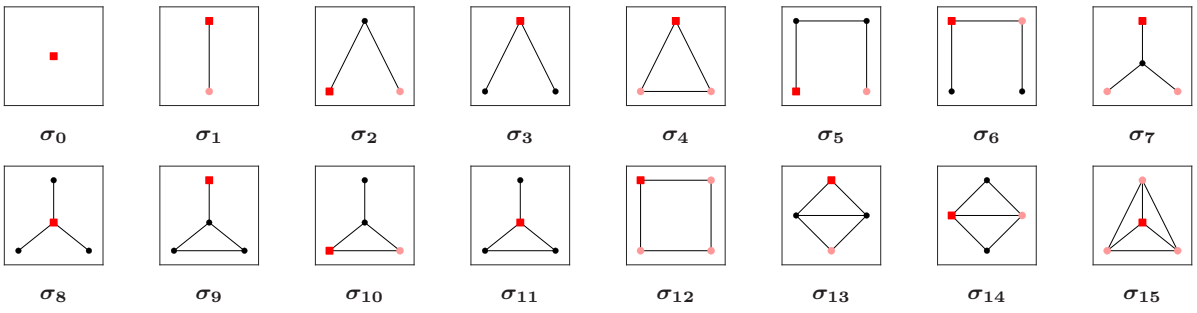

Figure 1: Dictionary Σ_{16} of 16 graphlets. In each graphlet, the designated incidence node is specified by the red square marker, its automorphic position(s) specified by red circles. The total ordering (labeling) of the graphlets is by the following nesting conditions. The graphlets are ordered first by non-decreasing number of vertices. Graphlets with the same vertex set belong to the same family. Within each family, the ordering is by non-decreasing number of edges, and then by increasing degree at the incidence node (except the 4-cycle). The inclusion of σ_0 is necessary to certain vertex partition analysis [4].

the same time, in ready form for globally streamlined computation in matrix-vector operations. (2) It leverages prevalent multi-core processors, with multi-threaded programming in `Cilk` [2], and uses sparse graph computation techniques to deliver high-performance network analysis to individual laptops or desktop computers. (3) It has `Python`, `Julia`, and `MATLAB` interfaces for easy integration with, and extension of, existing network analysis software.

2 Statement of need

With continuous and rapid growth in interest, understanding, and applications of the graphlet transform [4, 5, 8, 9, 10, 11, 13, 17], an efficient software for the transform is in great need, especially for advanced study of, and discovery in, diverse biological networks and emerging networks in many application domains.

With ease of use, the FG$_\ell$T user gets high-performance graphlet transform by several advanced features of the library. Instead of neighbor-by-neighbor search and recognition of induced subgraphs with the graphlet patterns, we use sparse and global formulas to theoretically and effectively reduce the redundancy among neighborhoods and make the computation streamlined in matrix-vector operations. We provide in Table 1 a summary of the formulas from [6]. We use multi-threaded programming to leverage multi-core processors in personal computers. We use sparse computation techniques, operation prefiltering and scheduling in order to reduce space complexity, the amount of unnecessary computation, the amount of data revisits, memory access latency, and imbalance among multi-threaded computation.

We illustrate in Table 2 the timing results with 12 networks of various types and sizes. All the experiments are on a single multi-core processor. The transform of network `com-LiveJournal` with 35 millions of edges takes less than 1 minute. The transform of network `com-Friendster` with 1.8 billions of edges takes about 2.5 hours with 16 threads while it takes 1.5 day with a single thread. The sequential computation alone substantially outpaces other available software.

References

[1] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[2] R. D. Blumofe, C. F. Joerg, B. C. Kuszmaul, C. E. Leiserson, K. H. Randall, and Y. Zhou, "Cilk: An efficient multithreaded runtime system," *Journal of Parallel and Distributed Computing*, vol. 37, no. 1, pp. 55–69, 1996.

[3] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 1082–1090.

[4] D. Floros, T. Liu, N. P. Pitsianis, and X. Sun, "Measures of discrepancy between network cluster configurations using graphlet spectrograms," 2020, manuscript under review.

Table 1: Sparse and exact formulas for the Fast Graphlet Transform (FG$_\ell$T) with dictionary Σ_{16}. There are two sets of formulas. The first set is for computing the raw frequencies $\hat{d}_j$, $0 \le j \le 15$, i.e., the frequencies of subgraphs with the graphlet patterns in dictionary Σ_{16}. The formulas are tabulated in the two tables to the left. For two vectors a and b, $a - b$ denotes the sparse/rectified difference $\max\{b - a, 0\}$. The difference between two sparse matrices is similarly denoted. The auxiliary formulas are in the bottom table. The second set of formulas is for converting the raw frequencies to the net frequencies d_j, $0 \le j \le 15$, i.e., the frequencies of induced subgraphs with the graphlet patterns. The matrix for forward conversion, from the net frequencies to the raw ones, is $U_{16} = U_5 \oplus U_{11}$. The matrix is shown to the right. The backward conversion matrix U_{16}^{-1} is explicitly formed and used, it has the same nonzero pattern as U_{16}.

Σ_{16}		Graphlet, incidence node	Formula in vector expression
	σ_0	singleton	$\hat{d}_0 = e$
	σ_1	1-path, at an end	$\hat{d}_1 = p_1$
	σ_2	2-path, at an end	$\hat{d}_2 = p_2$
	σ_3	bi-fork, at the root	$\hat{d}_3 = p_1 \odot (p_1 - 1)/2$
	σ_4	3-clique, at any node	$\hat{d}_4 = c_3$
	σ_5	3-path, at an end	$\hat{d}_5 = p_3$
	σ_6	3-path, at an interior node	$\hat{d}_6 = p_2 \odot (p_1 - 1) - 2\,c_3$
	σ_7	claw, at a leaf	$\hat{d}_7 = A\,\big((p_1 - 1) \odot (p_1 - 2)\big)/2$
	σ_8	claw, at the root	$\hat{d}_8 = p_1 \odot (p_1 - 1) \odot (p_1 - 2)/6$
	σ_9	paw, at the handle tip	$\hat{d}_9 = A\,c_3 - 2\,c_3$
	σ_{10}	paw, at a base node	$\hat{d}_{10} = C_3\,(p_1 - 2)$
	σ_{11}	paw, at the center	$\hat{d}_{11} = (p_1 - 2) \odot c_3$
	σ_{12}	4-cycle, at any node	$\hat{d}_{12} = c_4$
	σ_{13}	diamond, at an off-cord node	$\hat{d}_{13} = D_{4,c}\,e/2$
	σ_{14}	diamond, at an on-cord node	$\hat{d}_{14} = D_{4,3}\,e/2$
	σ_{15}	4-clique, at any node	$\hat{d}_{15} = T\,e/6$

U_5	d_0	d_1	d_2	d_3	d_4
$\hat{d}_0$	1				
$\hat{d}_1$		1			
$\hat{d}_2$			1		2
$\hat{d}_3$				1	1
$\hat{d}_4$					1

$$d_{0:4} = U_5^{-1}\,\hat{d}_{0:4}$$
$$d_{5:15} = U_{11}^{-1}\,\hat{d}_{5:15}$$

U_{11}	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
$\hat{d}_5$	1				2	1		2	4	2	6
$\hat{d}_6$		1				1	2	2	2	4	6
$\hat{d}_7$			1		1	1			2	1	3
$\hat{d}_8$				1			1			1	1
$\hat{d}_9$					1				2		3
$\hat{d}_{10}$						1			2	2	6
$\hat{d}_{11}$							1			2	3
$\hat{d}_{12}$								1	1	1	3
$\hat{d}_{13}$									1		3
$\hat{d}_{14}$										1	3
$\hat{d}_{15}$											1

Auxilliary formulas	
$P_2 = A^2 - \mathrm{diag}(p_1)$	$p_1 = A\,e$
$C_3 = A \odot A^2$	$p_2 = A\,p_1 - p_1$
$C_{4,2} = P_2 \odot (P_2 - 1)$	$p_3 = A\,p_2 - p_1 \odot (p_1 - 1) - 2\,c_3$
$D_{4,c} = A \odot (A\,(C_3 - A))$	$c_3 = C_3\,e/2$
$D_{4,3} = A \odot C_{4,2}$	$c_4 = C_{4,2}\,e/2$
$T = A \odot \lfloor q_{ij}^{\mathrm{T}}\,A\,q_{ij} \rfloor$, $q_{ij} = a_i \odot a_j$	

Table 2: Execution time of the Fast Graphlet Transform, with software library FG$_\ell$T in version 1.0.0, of 12 networks on a single Xeon processor. The networks are taken from two data sources (column 1), they are listed row by row in the table. Some are snapshots of real-world networks, the others are generated by models based on real-world networks. The basic statistics of each network are shown in columns 3 to 6: the number of nodes, number of edges, and average and maximal node degrees, where K stands for a thousand and M for a million. The execution time is wall-clock time in seconds (default), minutes and hours. It is reported under 5 execution options: single-thread (sequential), 2-threads, doubled up to 16-threads. Each execution time under a minute is the average over 5 runs. Speedup factors over the single-thread execution are placed next to the execution time. We observe that the speedup factors are lower with sparser networks. The experiments are carried out on an Intel Xeon E5-2640 v4, with 10 cores (up to 20 concurrent threads via hyper-threading) and 700 GB of DDR4 RAM. Cilk [2] is used for multi-threaded programming.

	Network $G=(V,E)$	$\|V\|$	$\|E\|$	d_{avg}	$d_{\max}$	FG$_\ell$T	Multi-threaded (with Cilk)							
						Seq.	2 thr.		4 thr.		8 thr.		16 thr.	
DIMACS10	smallworld [15]	100K	500K	10.0	17	0.3	0.2	1.8x	0.1	3.3x	0.1	5.3x	0.0	6.1x
	598a [14]	111K	742K	13.4	26	0.6	0.3	1.8x	0.2	3.3x	0.1	5.8x	0.1	7.0x
	preferentialAttachment [1]	100K	500K	10.0	983	0.8	0.4	1.9x	0.2	3.6x	0.1	6.5x	0.1	9.0x
	144 [14]	145K	1M	14.9	26	1.0	0.5	1.9x	0.3	3.3x	0.2	5.2x	0.1	7.0x
	vsp-model1 [12]	45K	190K	8.4	17.7K	5.3	2.7	2.0x	1.4	3.8x	0.8	6.9x	0.5	11.0x
	vsp-south31-slptsk [12]	40K	190K	9.6	17.7K	4.6	2.4	1.9x	1.2	3.8x	0.7	6.8x	0.4	11.3x
	coPapersDBLP [7]	540K	15M	56.4	3.3K	3m:35s	1m:51s	1.9x	58.1	3.7x	32.0	6.7x	18.1	11.9x
SNAP	com-Amazon [16]	335K	926K	5.5	549	1.0	0.6	1.7x	0.3	2.8x	0.2	4.0x	0.2	5.4x
	loc-Gowalla [3]	197K	950K	9.7	14.7K	13.8	6.9	2.0x	3.6	3.9x	2.0	6.8x	1.7	8.0x
	com-LiveJournal [16]	4M	35M	17.3	14.8K	8m:59s	4m:53s	1.8x	2m:48s	3.2x	1m:37s	5.6x	56.9	9.5x
	com-Orkut [16]	3M	117M	76.3	33.3K	1h:27m	46m:44s	1.9x	28m:57s	3.0x	18m:29s	4.7x	11m:25s	7.7x
	com-Friendster [16]	66M	1.8B	55.1	5.2K	32h:29m	16h:22m	2.0x	7h:59m	4.1x	4h:13m	7.7x	2h:34m	12.7x

[5] ——, "Using graphlet spectrograms for temporal pattern analysis of virus-research collaboration networks," in *IEEE High Performance Extreme Computing Conference*, 2020.

[6] D. Floros, N. Pitsianis, and X. Sun, "Fast graphlet transform of sparse graphs," in *IEEE High Performance Extreme Computing Conference*, 2020.

[7] R. Geisberger, P. Sanders, and D. Schultes, "Better approximation of betweenness centrality," in *10th Workshop on Algorithm Engineering and Experiments*, J. I. Munro and D. Wagner, Eds., 2008, pp. 90–100.

[8] N. Przulj, D. G. Corneil, and I. Jurisica, "Modeling interactome: Scale-free or geometric?" *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, Dec. 2004.

[9] ——, "Efficient estimation of graphlet frequency distributions in protein-protein interaction networks," *Bioinformatics*, vol. 22, no. 8, pp. 974–980, Apr. 2006.

[10] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, Jan. 2007.

[11] N. Pržulj, Ed., *Analyzing Network Data in Biology and Medicine: An Interdisciplinary Textbook for Biological, Medical and Computational Scientists*. Cambridge: Cambridge University Press, 2019.

[12] I. Safro, P. Sanders, and C. Schulz, "Advanced Coarsening Schemes for Graph Partitioning," *Journal of Experimental Algorithmics*, vol. 19, pp. 1.1–1.24, 2015.

[13] A. Sarajlić, N. Malod-Dognin, Ö. N. Yaveroğlu, and N. Pržulj, "Graphlet-based characterization of directed networks," *Scientific Reports*, vol. 6, no. 1, p. 35098, Dec. 2016.

[14] A. Soper, C. Walshaw, and M. Cross, "A combined evolutionary search and multilevel optimisation approach to graph-partitioning," *Journal of Global Optimization*, vol. 29, no. 2, pp. 225–241, 2004.

[15] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[16] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[17] Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, and N. Pržulj, "Revealing the hidden language of complex networks," *Scientific Reports*, vol. 4, no. 1, p. 4547, May 2015.