Network Motif & Triad Significance Profile Analyses On Software System

Zhang Lin, Qian GuanQun,Zhang Li
School of Compute Science and Engineering
Beihang University
37# Xueyuan Rd, Beijing
CHINA
zhanglin@cse.buaa.edu.cn,qianguanqun@cse.buaa.edu.cn,lily@buaa.edu.cn

Abstract There has been considerable recent interest in network motif for understanding network local features, as well as the growth and evolution mechanisms. In order to discover the local features in software networks, we extended the network motif research methods to software domain. After comparing triad significance profiles from 138 java open source software packages, we found three typical kinds of network motifs. Moreover, the software networks could be divided into 3 clusters which are consistent with the known super-families from various other types of networks. It seems that software scale and interaction may be the reasons causing different motif SP distribution. The concepts, the principles and steps associated with the experiment were elaborated, as well as the results were analyzed and discussed, the direction for further research was given.

***Key Words**:* Software Network, Network Motif, Triad Significance Profile, Superfamily

1 Introduction

A lot of empirical study found that two basic features common to many complex networks, from the Internet to metabolic nets, are their scale-free (SF) [1] topology and a small-world (SW) [2] structure. The first states that the proportion of nodes P (k) having k links decays as a power law. Examples of SF nets include Internet topology, cellular networks, scientism collaborations [3] and lexical networks [4]. The second refers to a web exhibiting very small average path lengths between nodes along with a large clustering [1, 5]

Software is built up out of many interacting units and subsystems at many levels of granularity subroutines, classes, source files, libraries, etc, and the interactions and collaborations of those pieces can be used to define networks or graphs that form a skeletal description of a system.

Software systems, especially large-scale software systems, can be decomposed into a collection of entities such as class, components etc. If we regard entities as nodes, the interrelationship between the entities as edges, then the software system can be seen considered as a network which we called as software network. The software network share some similar global features with other complex network, such as small world, high clustering coefficient and power law distribution [6] [7-8]

The research shows that the complex network having similar global features may have a completely different local structure, which is important to understanding the growth and evolution mechanism of complex networks is of great significance. Therefore, the research for the local features deserves more attention.

Current research shows that the network motif contributes to the understanding the growth and evolution mechanism from the network local features. Network motif is higher frequency subgraph in a real network than in a random network [9]. Each network can be depicted as a specific set of network motifs which is the typical local pattern. At the same time, identify these network motifs facilitates the clustering of the network connection mode and helps understand the growth and evolution mechanism [10-11].

For example, the feedforword loop with three nodes is a typical local connection pattern in transcription and neural networks. The feedback loop with four nodes is a typical local connection mode in the electronic circuits rather than the biological network. The network motif in yeast protein network is the reflection of its high evolution and reflects the trend of the evolution of species. [10, 12]

Ron Milo and etc present an approach [13] to systematically study similarity in the local structure of networks, based on the significance profile (SP) of small subgraphs in the network compared to randomized networks. They find several superfamilies of previously unrelated networks with very similar SPs. One superfamily, including transcription networks of microorganisms, represents “rate-limited” information-processing networks strongly constrained by the response time of their components. A distinct superfamily includes protein signaling, developmental genetic networks, and neuronal wiring. Additional superfamilies include power grids, protein-structure networks and geometric networks, World Wide Web links and social networks, and word-adjacency networks from different languages. However, the software network not is included in their research and analyses.

Software network motif is a basic software module building the software network. Its frequent emergence may be a reflection of the internal collaboration and reuse pattern in software system. Software networks motif represent another important motif in complex networks, which to date have received relatively little attention in this field.

In this paper, we extend the research of Ron Milo and his colleagues to software networks and try to discuss the fellow questions: (1) what the network motif belong to software network; (2) What is the software network superfamily? (3) What other field network is similar with the software network.

The organization of this paper is as follows: In Section 1, we introduce the background. In section 2, we define the software network model and how to get it. Section 3 presents the conception and principle of SP approach. Section 4 presents the empirical design and process. In Section 5, we present an analysis to the empirical result. Finally, we draw the conclusion in section 6 and give the future research work and challenges.

2 The Definition of Software Network

In object-oriented programming language, the class encapsulates the data and the behavior as well as the interface. Regard the classes as the nodes and the relationship between the classes as the edges, the software system just presents a network topology which reflects the internal relationship of interaction and collaboration.

We focused on object-oriented Java software and mapped each class or interface to a network node. The primitive data type, such as char, long, double and etc was ignored. The following dependency relationships between classes were considered. [14]

(1) If class A extends class B, a link from A points to B.
(2) If class A implements interface B, a link goes from A to B;
(3) If class A contains a member whose type is class B, a link from A points to B;
(4) If class B is passed into a member function of class A as a parameter, a link exists between A and B.
(5) If the implementation body of a member function of class A contains a variable whose type is class B, a link from A points B.

Figure 1 shows a demonstration of the Java network’s definition.

```
Public class Painter extends Person implements
Paint-Skill
{
    Private Address adds;

    public void paint (Furniture fern)
    {
        Logger log;
        // paint furniture and save log info;
    }
}
```

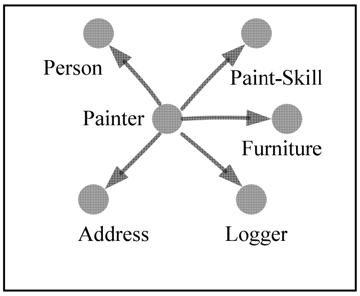

Fig.1 A sample of Java code and corresponding directed network (all 5 cases of dependency relationships are considered)

3 Principle of the approach

3.1 Concept & Principle

3.1.1 Network motif

Although the shape of network is diverse and complex, the mode and process of building network has certain disciplines. Recent research presents that the connection pattern is not of random, some of which is typical connection. In other word, the typical connection is repeating. The number which they appeared in a real network is far higher than in the random network. Milo etc called the above connection patter as network motif.

Network motif is different in distinct network, which is considered the basic elementary building blocks. The large-scale networks are built by repeating copy and reuse the motif.

3.1.2 Triad Node Connected graph

The triad census of a given network consists of 16 values for the number of appearances of each of the possible 3 node subgraph in directed networks, 13 of these 16 subgraphs are connected. (See Fig 2)

The four-node subgraph of a given nondirected networks consists of 199 values for the number of appearances of each of the possible 4 node subgraph. 3-node subgraph is the basic building blocks of local structure of any complex network. In a software network, the single and mutual edges in these subgraphs show collaborative relationships between software components. Design patterns are good examples, also with basic topology structures that are 3-node subgraphs.In this paper, only the triad node connected graph is consider.

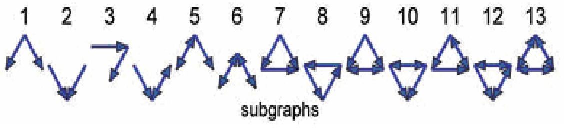

Fig.2 13 kinds of the connected component

3.1.3 The sample of Detection in network motif

To focus on those that are likely to be important, we compared the real network to suitably randomized networks and only selected patterns appearing in the real network at numbers significantly higher than those in the randomized networks. For a stringent comparison, we used randomized networks that have the same single-node characteristics as does the real network: Each node in the randomized networks has the same number of incoming and outgoing edges as the corresponding node has in the real network. The comparison to this randomized ensemble accounts for patterns that appear only because of the single-node characteristics of the network (e.g., the presence of nodes with a large number of edges). Furthermore, the randomized networks used to calculate the significance of *n* node subgraph were generated to preserve the same number of appearances of all (*n* – 1)-node subgraph as in the real network. This ensures that a high significance was not assigned to a pattern only because it has a highly significant subpattern. The "network motifs" are those patterns for which the probability *P* of appearing in a randomized network an equal or greater number of times than in the real network is lower than a cutoff value (here *P* _ 0.01)[9] .

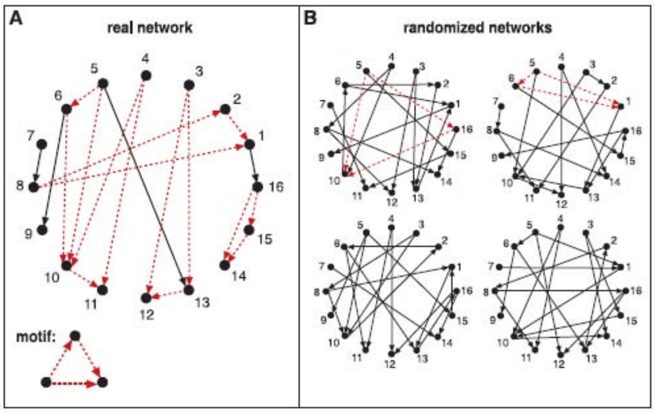

*Fig3***.** Schematic view of network motif detection. Network motifs are patterns that recur much more frequently (**A**) in the real network than (**B**) in an ensemble of randomized networks. Each node in the randomized networks has the same number of incoming and outgoing edges as does the corresponding node in the real network. Red dashed lines indicate edges that participate in the feedforward loop motif, which occurs five times in the real network.

Schematic view of network motif detection see the Fig 3.In the Fig, Network motifs are patterns that recur much more frequently (**A**) in the real network than (**B**) in an ensemble of randomized networks. Each node in the randomized networks has the same number of incoming and outgoing edges as does the corresponding node in the real network. Red dashed lines indicate edges that participate in the feedforward loop motif, which occurs five times in the real network and occurs two times in four random networks.

3.2 Determine Triad Signification Profile method

Through the example above, we can see the present method of detecting network motifs is based on contrasting the number of appearances of subgraph in a network to an ensemble of random networks.

3.2.1 Methods for generating randomized networks with a given degree sequence:

The random networks used here conserve the degree sequence of the original network for incoming edges, outgoing edges and mutual edges.
A pair of edges is randomly chosen and switched (A->B, C->D becomes A->D, C->B). This is repeated for a predetermined number of trials T. If the switch generates multiple edges or self-edges, or makes no change in the network, it is not performed (but is counted as a trial in order to ensure detailed balance).So the switched method ensures to generate random graphs with a given degree distribution [13]
To ensure the accuracy of statistic in empirical results, the number of random network at least should be 1000.

3.2.2 The Motif Detection Algorithm for Significance Profile

The motif detection algorithm for subgraph significance profile is contrasting the number of subgraph appears between in the real network and in the randomized network which has the same nodes and degrees.

The network motifs are those patterns for which the probability *P* of appearing in a real network an equal or greater number of times than in the randomized network is higher.

To ensure the accuracy of statistic in empirical results, the number of randomized network at least should be 1000.

The motif detection algorithm is based on a hypothesis test. The null hypothesis is that the number of subgraph appears has no significant difference between the real network and the randomized network. On the contrast, the alternate hypothesis is that the number of subgraph appears has no significant difference.

So, for each subgraph *i*, the statistical significance is described by the *Z* score

（1）

$$Zscore = \frac{N_{real} - < N_{rand} >}{std(< N_{rand} >)} \tag{1}$$

$$< N_{rand} >= \frac{\sum_{i=1}^{n} N_{i\_rand}}{n} \tag{2}$$

$$std(< N_{rand} >) = \sqrt{\sum_{i=1}^{n} \frac{(N_{i\_rand} - < N_{rand} >)^2}{n-1}} \tag{3}$$

Where Nreal is the number of times the subgraph appears in the network, and N_{randi} and STD (Nr_{andi}) are the mean and standard deviation of its appearances in the randomized network ensemble.

（2）P value

$$P = \frac{\sum_{i=1}^{n} p_i}{n} \tag{4}$$

$$p_i = \begin{cases} 1 & if \quad N_{i\_rand} > Nreal \\ 0 & otherwise \end{cases} \tag{5}$$

P value used to measure the inconsistent deviation between the sample values with null hypothesis. The range of P value is 0 to 1...The smaller P values are, the more sufficient the preseason to rejecting null hypothesis are.

（3）SP value

The SP is the vector of Z scores normalized to length 1:

$$SP_i = \frac{Z_i}{\sqrt{\sum Z_i^2}} \tag{6}$$

The normalization emphasizes the relative significance of subgraph, rather than the absolute significance. This is important for comparison of networks of different sizes, because motifs (subgraph that occur much more often than expected at random) in large networks tend to display higher Z scores than motifs in small networks [13].

3.2.3 Criteria for Network Motif Selection

For the purposes of the present study, network motifs are subgraph that meets the following criteria:

(1)The probability that it appears in a randomized network an equal or greater number of times than in the real network is smaller than $P = 0.01$. In the present study, P was estimated (or bounded) by using 1000 randomized networks.

(2) The number of times it appears in the real network with distinct sets of nodes is at least $U = 4$.

(3) The number of appearances in the real network is significantly larger than in the randomized networks: $N_{\text{real}} - N_{\text{rand}} > 0.1N_{\text{rand}}$. This is done to avoid detecting as motifs some common subgraph that have only a slight difference between N_{rand} and N_{real} but have a narrow distribution in the randomized networks.

3.3 Clustering for Super-families

The similarities between networks can also be visualized by looking at the correlation between the TSPs of different networks. The correlations can be used to cluster the networks into distinct superfamilies. The TSPs display certain conservation relations between subgraph types. For example, networks with excess triangle-shaped triads tend to have a deficit of V-shaped triads. There are several triad conservation rules in networks that conserve the degree sequences of single and mutual edges . As a result, the 13 values of the TSP are determined by only seven degrees of freedom. One can intuitively interpret these conservation laws in terms of reactions that convert V-shaped subgraphs into triangles, preserving the degrees of all participating nodes. Analysis of the reactions occurring in each network allows a compact description of the difference between networks and their randomized counterparts.

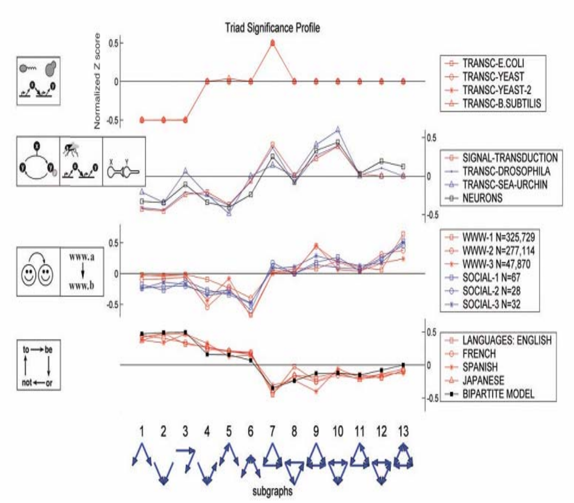

Fig.4 The triad significance profile (TSP) of networks from various disciplines. The TSP shows the normalized significance level (Z score) for each of the 13 triads. Networks with similar characteristic profiles are grouped into superfamilies

4 Software motifs TSP experiment

According the above principle, we design the experiment to detect the software network motifs and clustering the software network.

The experiment process as fellow:

(1) Select the sample data

The experiment data come from 138 java open source software packages (.jar packages).

These softwares have various sizes. The number of nodes range from several (small size) to several hundred (medium size) even to several thousands (large size).

(2) Build software network

We use DependencyFinder [15] to extract dependency relationships between classes in Java .jar packages and translate the classes and relationships into a network.

(3) Detect the software network motifs

We used Mfinder [16] to calculate the Z score and P value. All the subgraphs which meet the criteria (see the section 3.2.3) are software network motifs,

(4)TSP Trend Analysis

We normalized the Z score to SP value. These SP values (13) reflect the trend of each subgraph SP.

(5) Clustering analysis

Detect which network is TSP trendy most similar the software network.

All results in the present paper were obtained from the follow URL:

http://groups.google.com/group/complex-network-group-buaa

5 Experiment Result Analysis

5.1 What are the software motifs

We get the 3 most significant network motifs from statistic data come from 138 java software systems. They are the No.7, No.10 and No.9 in the Fig2, see the Table 1.

The three connected graph detected to be a motif are of very high proportion, respectively 96.26%, 73.83 %and 51.40%. In other word these three graphs appear to be higher frequency in real network than in random network. This means they represent some common design pattern or be used to perform certain specific tasks and functions.

Among them, the 7th of connected subgraph is considered as motif in a highest proportion. It just reflects the commonly design pattern, When the Class A interact with Class B, they used to call another functions in class C. The typical design pattern such as proxy can be mapped into the 7th connective graph, and many of other design patterns can be mapped into the variants of the 7th graphic, shown in the Fig 5(b), (d) and (d). From the above analysis we can see that the motifs in the software network which the frequency is much higher than in the random network are caused by the design of mechanisms.

Table 1 The typical triad motif in software

Subgraph	Proportion	Samples in the real network					
		Software Name	Number of Node	Number of edge	N_{real}	N_{rand} （std）	Z_{score}
 7th	96.26%	Emf	89	264	234	133.3 (11.2)	8.96
		Gef	531	1770	1022	436.6 (34.0)	17.23
		Ant	752	3306	3588	2057.5 (99.0)	15.46
 10th	73.83%	Ant	752	3306	252	130.2 (11.8)	10.33
		Icu	703	1970	374	83.5 (10.7)	27.08
		compiler	89	475	57	27.8 (5.8)	5.08
 9th	51.40%	Ant	752	3306	385	108.4 (40.5)	6.84
		javasvn	378	2250	139	18.9 (7.2)	16.64
		Compare	378	1365	124	22.9 (8.4)	12.1

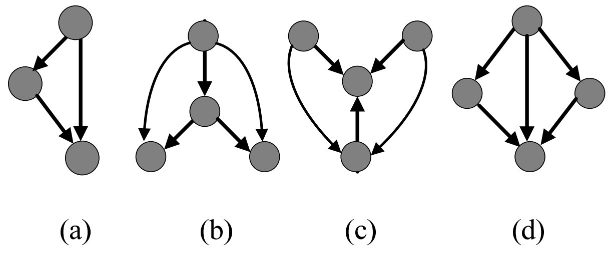

Fig.5 the 7th graphic and duplicate itself into four-node graphic.(a) the 7th graphic;(b),(c),(d) the 7th graphic duplicates itself into four-node graphic.

In particular, the 7th motif is also the gene network motifs, neural networks, as well as biological network motifs. Whether the similar evolution mechanism between the network motifs in various fields exists deserves future research.

In the study, we found another interesting phenomenon that the complex subgraphs which have melti-mutual edges (as seen from Fig2 No.12, No.13) almost no significant difference than in random network. It just reflect the software design methology which avoid the loop circuit to the greatest extent ,otherwise increases the complex of systemic and understanding and reduce the software structural stability

Since a network representation of a system usually does not represent all of the details of each interaction, it is not always possible to understand the function or dynamics of a given structure without additional information. For the reasons, the network motifs discovered lack of the clear semantic meaning for software system. So we can not be defined the uniqueness of the corresponding with functions and structure. Networks motifs reflect the limitations and character in system which is consistent with the latest research results by Ingram [17]

From what has been discussed above, various software systems which reflect the different of design pattern and organization behavior map different networks. The topology of motif is related to the scale of software and design principle.

5.2 Calculation of the correlation coefficient clustering the software network

To calculate the correlation coefficient between networks we used the Pearson Correlation Coefficient of the 13 profiles. The correlation matrix for non-directed networks at the level of tetrads is shown in Fig. 6. One can use clustering methods to group 138 different software networks according to the correlations in their TSP profiles.

As the result has been seen in the Fig 6.The software network can be dividing into 3 classes according to the correlations in their TSP profiles. Matrix

Figure6 is serialized result of correlation coefficient matrix of SP by Permut Matrix [21]. We found the software network can be roughly divided into 3 categories based on SP. The top left corner of the matrix is cluster1 (C1), Central block is cluster2 (C2), and bottom right is cluster3 (C3). The proportion is about 3.6%, 78.9% and 17.3%.

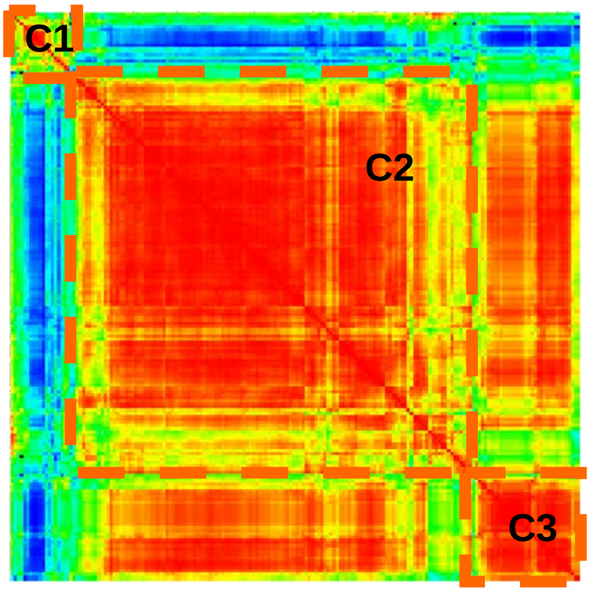

Fig.6 The correlation coefficient matrix of TSP.

5.3 TSP Trend Analysis

Corresponding to the result of clustering analysis, we plot the SP values of 13 sub-graphs for each cluster in Figure 3.C1, C2 and C3. Their distributions are similar with ones of superfamilies from previous research (shown in Figure 7.S1, S2 and S3). S1 come from Language network, S2 come from signal-Transcution, and S3 come from Transc-YEAST &ECOU.

Here are some interesting observations:

1. In most software (about 96%), Sub-graph7 is the most important motif and may present some basic design principle. For example, when class A collaborates with class B, they usually would share some sub functions.
2. C1 is totally different than C2 or C3. It seems to be the reverse. In C1, sub-graph 1, 2 and 3 are encouraged. C2 and C3 both have high positive value on sub-graph7 and low negative one on sub-graph1, 2 and 3. But SP values on sub-graph 9 and 10 in C2 are much higher than in C3.

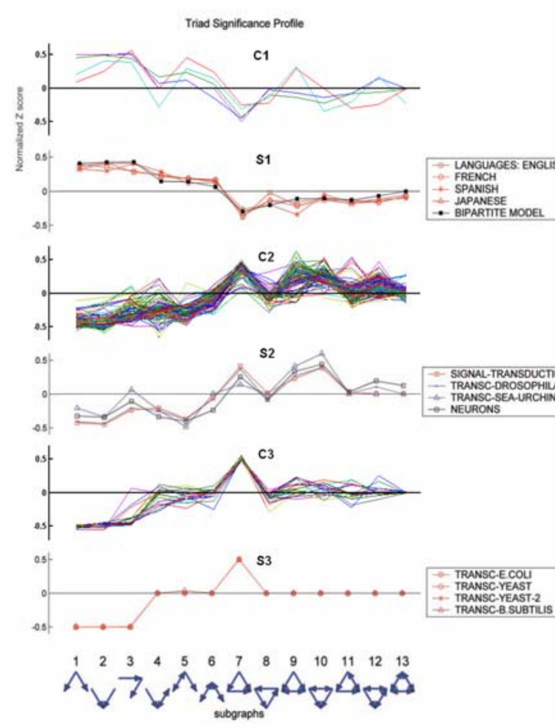

Fig.7 software network TSP trend Comparison Chart.C1, C2 and C3 from software network are respectively similar to S1, S2 and S3 which are come from the research result by Ron Milo .etc

Therefore, we may safely draw the conclusion that the same motif should be in the same type network. Moreover, the relative significance of motifs is similar**.**

The three kinds of software network are analyzed and compared in the global features and are shown in the Table 2.

Global statistic properties of each cluster is shown in Table 2, where <N>, <L> and <C> are the mean of class number, relationship number and clustering coefficient of networks in same cluster.

Through comparing part of the global features of each cluster, we found that the distribution of SP value seems to be related to software scale and collaboration:

(1) The scale of C1 is smallest in the three networks. Its mean number of nodes is only 34.2，the mean number of edges is 66.4，and the mean clustering coefficient is 0.089. This means the collaboration and interaction are sparse. As well as the motif of C1 is 1st, 2nd and 3rd subgraph. We infer from above discussion, the topology of C1 is star or tree structure and organic level is ordering.

(2) The scale of C3 is medium in the three networks. Its mean number of nodes is 71, the mean number of edges is 257.1 and the mean clustering coefficient is 0.208.This means that the adding interaction with scale of network. The topology of C3 is net structure because the 7th subgraph is the motif in this kind of network.

(3) The scale of C2 is largest. Its mean number of node is 351.1, the mean number of edges is 1129 and the mean clustering coefficient is 0.230.This means that TSP raised with 7th, 10th and 9th subgraph becomes the motif of network. The clustering coefficient limit increased because of the principle of reuse and design optimize

The clustering result was only statistically significant, mean nothing at all the network must be of some scale .In fact, the number of nodes in the C2 network span is very big ,from several dozens to several hundred. This will further enhance that different scale and style networks may be in the same superfamily.

Table 2 Global features of clusters based on the TSP

Network clustering	<N>	<L>	<C>	$\alpha = \left\langle \frac{\log L}{\log N} \right\rangle$
C1	34.2	66.4	0.089	0.764
C2	315.1	1129	0.230	1.204
C3	71	251.7	0.208	1.198

Here, according to the trendy of TSP, we divided the software into 3 classifications and it is helpful to attain an awareness of software system and network by contrast of the research of result. At the same time, the researches of other fields are bound to contribute to that of software. The success in other fields has a contributed a great deal to the growth and evolution mechanisms of software.

6 Conclusion and future work

Software Network has been testified to has the global properties of complex network, such as small world, scale free and modularity and hierarchy by the high coupling and low cohesion design principle.

Software systems represent another important class of complex networks, which to date has received relatively little attention in this field. Software network has not been further in-depth research contrast to the other fields (biological network, social network, WWW network etc), especially in local structure research.

We learn the method for detecting network motifs in other field and research on super-families from Moli .etc, building the software network model , detecting what are the motifs in software network and analysis the reasons. On the foundation, we made a TSP analysis and found that all these networks could be divided in 3 clusters. Which was respectively similar to LANGUAGE NETWORK, SIGNAL-TRANSCUTION, and TRANSC-YEAST &Couther further analyze show that the feature of TSP in each cluster seems to be relative to the scale of network.

These experiments make us get a clear version of software network. We will benefit from some methods come from other field which are in the same superfamily with software network, which help us to understanding the growth and evolution mechanisms of software networks.

Therefore, there are some well worth further reflecting and exploring problems. What are the internal reasons lead to TSP similar with in same clustering network? How are the differences of TSP in distinct software clustering? Whether the TSP difference has any relevance for adaptability, program language, design pattern or scale of software? Moreover, is there any relation between the global features with local features? Could they are predicting each other? All these problems are worth further studying and discussing to us.

7 Acknowledgments

This research project was supported by the National Natural Science Foundation of China under Grant No.60773155 and the Key Basic Research Developing Project (973 project) from Science and Technology Ministry of China under Grant No.2007CB310800.

References:

[1] R. Dobrescu, "Modeling complex biological systems using scale free networks," presented at Proceeding of the 5th WSEAS Internal Conference on Non-Linear Analysis,Non-Linear Systems and Chaos, 2006.

[2] L. A. N. Amaral, A. Scala, MBarthélémy, and H. E. Stanley, "Classes of small-world networks," *PNAS*, vol. 97, pp. 11149-11152, 2000.

[3] M. E. Newman, "Scientific collaboration networks. I. Network construction and fundamental results," *Physical Review E*, vol. 64, pp. 016131, 2001.

[4] P. H. YIHJIA TSAI, "A Growing Network Model with High Hub Connectivity and Tunable Clustering Coefficient," *WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS*, vol. Volume 4, pp. 1447-1452, 2007.

[5] S. T. S. Mocanu, "Immunization strategies for networks with scale-free topology," presented at Proceedings of the 5th WSEAS International Conference on Non-Linear Analysis, Non-Linear Systems and Chaos, 2006.

[6] M. A. de, Y. C. Lai, and A. E. Motter, "Signatures of

small-world and scale-free properties in large computer programs," *Physical Review E*, vol. 68, pp. 017102, 2003.

[7] D. Hyland-Wood, D. Carrington, and S. Kaplan, "Scale-Free Nature of Java Software Package, Class and Method Collaboration Graphs," MIND Laboratory,University of Maryland College Park. Tech. Rep. No. TR-MS1286, 2006.

[8] R. Wheeldon and S. Counsell, "Power law distributions in class relationships," presented at Source Code Analysis and Manipulation, 2003. Proceedings. Third IEEE International Workshop on, 2003.

[9] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network Motifs: Simple Building Blocks of Complex Networks," *Science*, vol. 298, pp. 824-827, 2002.

[10] S. Maslov, K. Sneppen, and A. Zaliznyak, "Detection of topological patterns in complex networks: correlation profile of the internet," *Physica A: Statistical and Theoretical Physics*, vol. 333, pp. 529-540, 2004.

[11] J. Qin, S. Pinkenburg, and W. Rosenstiel, "Parallel motif search using ParSeq," presented at IASTED International Conference on Parallel and Distributed Computing and Networks, as part of the 23rd IASTED International Multi-Conference on Applied Informatics, Feb 15-17 2005, Innsbruck, Austria, 2005.

[12] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of Escherichia coli," *Nat Genet*, vol. 31, pp. 64-68, 2002.

[13] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of Evolved and Designed Networks," *Science*, vol. 303, pp. 1538-1542, 2004.

[14] QianGuanQun and Z. ZhangLi, "Modeling Method and Characteristics Analysis of Software Dependency Networks," *comper science*, 2008.11

[15] DependencyFinder, , 1.2.0 ed: http://depfind.sourceforge.net/, 2007.

[16] Mfinder, , 1.2 ed: http://www.weizmann.ac.il/mcb/UriAlon/groupNetworkMotifSW.html.

[17] M. P. S. A. J. S. Piers J Ingram , "Network motifs: structure does not determine function," *BMC Genomics*, vol. 7, pp. 108, 2006.