Prof. Alfredo Ferro
V.la A. Doria, 6
I-95125 Catania ITALY
e-mail: ferro@dmi.unict.it

November 30, 2009

Dear Prof. Jack Cochrane,

on the behalf of the other co-authors enclosed you will find our revised manuscript "SING.pdf" titled "SING: Subgraph search In Non-homogeneous Graphs". We very much appreciated the reviewers' comments and modified our manuscript accordingly. A list of all modifications according to the reviewers suggestions follows. Please let us know if everything is in order.

Best wishes,

Alfredo Ferro

Reviewer 1

1. Although the authors claim to solve the problem in the context of large graphs, almost all of the empirical evaluation focuses on the databases of small graphs. Only a paragraph is devoted to evaluate the efficiency of the proposed technique.

We added a new subsection describing evaluation on a PPI network.

Therefore the following questions need to be answered:

a. Why is the large graph synthetically generated when so many large graphs naturally occur (such as social networks, large protein networks)? If social networks are too large for the proposed technique, then what are scalability boundaries in terms of graph size and density?

We applied the tool to the PPI network of human for finding

yeast complexes.

b. What is the degree distribution in the graph? If the distribution is uniform, then a degree of 2 is too low to conclude anything. Large graphs of such low degree is extremely rare.

The following text was added:

The network was generated by adding edges one at the time in the following way. Each new edge is connected to an existing node of the network with probability proportional to the degree of that node. This procedure produces a network with power law degree distribution.

The degree distribution of the PPI network is described in Figure 7b

c.What is the distribution of the different node labels?

We added this sentence to the paragraph of synthetic graph:

"The labels were assigned by distributing 8 different labels randomly (with uniform distribution) over the network nodes."

Moreover the distribution of labels in the PPI network is described in Figure 11.

d.How were the query graphs selected? What were the sizes of the query graphs?

We added the following statement

"We queried the graphs with three different sets of 10 queries having size (number of edges) 4, 8 and 16 respectively. These queries were randomly generated using the same procedure of the "Molecular data" section".

e.Why is CTree used for comparison instead of a feature based technique? Ctree will be least efficient when there is a single graph since the index will be the graph itself.

We removed the comparison with CTree. Moreover, comparisons with other feature based tools (gIndex, Tree+Delta) are not re-

ported since they do not perform on large graphs. Indeed they were not designed for large graphs

2. The experiments either ignore or make improper use of feature based graph indexing techniques. In the comparison with gIndex, why do the authors force gIndex to have an index size similar as theirs? A larger index does not necessarily mean better query processing time. gIndex uses a discriminative ratio threshold to decide which features to index, and forcing a fixed index size would make this parameter inefficient. Moreover, a larger index means more query processing time. Therefore, no conclusion can be drawn from the running time plot in Figure 5. The candidate size plot in Figure 5 indicates gIndex as more efficient.

Other experiments using different parameters of Gindex were performed and illustrated (see fig. 5).

3. The core contribution lies in extending the existing GraphGrep algorithm. Instead of only indexing paths, the authors also track the starting node of each path. Although the following extension captures more structural information, it is fairly straightforward and does not offer any significant advantage over indexing trees. The only advantage of the indexing paths could be faster query running time. However, this crucial comparison is missing in the empirical section.

We added the following statement to the "Results and Discussion" section :

Dealing with large graphs, in some cases Treepi approach is computationally expensive. Indeed, enumerating all trees produces an explosion of the number of features that must be reduced by a data mining step. This leads to increase the preprocessing time as well as keep the filtering performances limited, due to the small number of features selected. Moreover, Treepi requires the computation of all pairwise distances between features. To bound the preprocessing and filtering time, a small number of features needs to be selected. Consequently, an high number of candidates is produced. On the other hand, SING considers only all paths starting from a node. This requires much less computation yielding lower preprocessing and filtering time. Furthermore, SING is able to capture the topology of the tree induced by a node, us-

ing simple paths. Consequently, it requires a smaller number of features avoiding the expensive feature selection process.

Finally, we compared our tool with Tree+Delta, a tree-based approach.

4. The author maintain that gIndex was unable to run on AIDS dataset with the inclusion of H atoms. How large were the graph with H atoms added? Without H atoms the graphs have an average of 26 nodes and 27 edges. Even if they triple in size (considering at most 3 H atoms can be added to C atoms which comprises the majority of the nodes), the graphs would still be within 100 nodes. Therefore it appears almost all graphs would satisfy the constraint of 250 nodes and 250 edges. The authors should mention how many graphs were filtered out due to this constraint.

43 graphs were discarded from the whole database. Moreover gIndex was not able to run on the resulting dataset, so a small dataset of 8000 graph was considered. The following sentence was modified in the paper:

From the whole AIDS compounds database, 8000 graphs having less than 250 nodes and less than 250 edges were selected.

Changed in:

From the whole AIDS compounds database, 43 graphs having more than 250 nodes or edges were first discarded. Then, 8000 graphs were randomly selected from the resulting dataset.

Minor Essential Revisions: 1. Why is the proposed technique referred to as GraphFind in most of the plots? GraphFind is a completely different work.

That was a misprint that we have corrected.

2. Page4: "to find" should be "to finding"

Addressed

Reviewer 2

In Figure 2-, "GraphFind" should be SING to be compared. Since GraphFind was reported, it needs to clarify it if SING is built on top of GraphFind. In Figure 2, the candidates after pruning are given.

That was another misprint that we have corrected.

It seems that CTree outperforms the others. It needs to discuss it more, since the query processing time depends on the number of candidates.

We argued that CTree is less efficient by the following statement:

"In more in detail, CTree uses an expensive filtering process based on an approximated algorithm for subgraph isomorphism. As a matching algorithm it uses a variant of the Ullmann algorithm [?] integrated in the framework, which has been shown to be outperformed by VF2."

Review all Figure numbers inside the paper

Figure number were reviewed