In this section, we evaluate our algorithms on two datasets, a set of synthetic graphs and a set of graphs modeling active ingredients in antiviral drugs for the treatment of AIDS. We call the former the synthetic graph dataset and the later the real graph dataset. We perform two sets of tests; In the first set, we evaluate our algorithms on the retrieval of induced subgraph to query and In the second set, we evaluate retrieval of non induced sugraph to query. To test the efficacy of our algorithm with regard to induced subgraph query, we evaluate the proposed algorithm in comparison with Messmer's Network Algorithm and a sequential SCAN using the state-of-the-art subgraph isomorphism detection algorithm VF2cordella2001$_v f2. Wealsoevaluateouralgorithmwithregardtonon-inducedsubgraphquery, wherewecompareproposed$
$+ programminglanguageandrunonaIntelCorei-3CPU3.09GHZ, 16Gbytememory, PersonalComputerrunningWindow$

Graph Datasets We evaluate our algorithms by processing the retrieval of descriptors from the compounds dataset. We use AIDS Antiviral Screen dataset to provide a real graph dataset. This dataset contains around 43,000 chemical compounds and is available publicly from NCI National Cancer Institute http://dtp.nci.nih.gov/ We denote this dataset as "AIDS" in our experiment. The graphs of AIDS have an average number of 25 vertices and 27 edges, a maximum of 438 vertices and 441 edges, 63 distinct vertex labels and 3 distinct edge labels.

In order to evaluate our algorithms over a larger database to test scalability, we use synthetic large dataset. The graph generator is configured to emit only connected graphs thathave an edge probability of 50 percent. Except where explicitly stated, 10 distinct vertex labels and 10 distinct.

figure[h] [width=1.0]images/realdataplot.pdf Results of the Experiments on the Real World Graph Data Set

Top left:Linear-linear plot of induced subgraph query processing time for increasing size of query graph.The smallest query graph contains 10 vertices while the largest query graph contains 60 vertices. The Fast Network Method and the textitNetwork Method show almost identical processing times. The VF2 method takes more than 50s to process query graphs large than 10 vertices.

Top Right: Linear-log plot of the induce subgraph graph query processing time with query graph size. We observe that both the Fast Network Method and the textitNetwork Method while almost indistinguishable, show an order of magnitude advantage in processing time over the state-of-the-art VF2 Method

Bottom left:Linear-linear plot of query processing time for increasing size of graph query.The smallest query graph contains 10 vertices while the largest query graph contains 60 vertices. The Fast Network Method is faster than VF2 for subgraphs too. Here too, VF2 method takes more than 50s to process query graphs large than 10 vertices.

Bottom Right: Linear-log plot of the graph query processing time with query graph size. We observe that both the Fast Network Method and the textitNetwork Method while almost indistinguishable, show better than an order of magnitude advantage in processing time over the state-of-the-art VF2 Method.

fig:fig81

p://dtp.nci.nih.gov/ .