Estimation of Motif Frequency in a Dynamic Network Topology

Pramit Dutta – 75132433 ; Kingshuk Mukherjee – 92820942 ; Bijoykrishna Saha - 93691593

Abstract:

A motif is a specific pattern of statistical and functional significance that repeats several times in a network. Motifs thus have a great importance in analysis of biological networks. A biological network evolves or changes very fast owing to mutations. Keeping a track of the dynamic changes that happen over a network is a challenging due to the vastness of a biological network. Algorithms exist that calculates the frequency of motifs in a static biological network. In this project we aim to address the dynamic behaviour of the network and come up with an algorithm that calculates or estimates the frequency measures in least possible time without running the static algorithms on the static instance of a dynamic network.

Background:

There are many approaches for determining the frequency of a motif in a static network. But all these methods are computationally intensive. When the topology of the network changes by a small margin, it becomes wasteful if we have to again compute the motif frequencies in the new network from scratch. In this project, we have attempted to design a method to estimate the motif frequencies in the changed (modified) network by updating the frequencies from the initial static network.

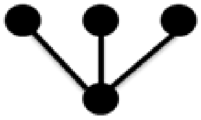
Concepts for the determination of pattern frequencies

There are 3 different measures for the determination of pattern frequencies in a network. [1] The first measure is called the F1 measure. This measure does not have any constraints in terms of overlaps and is simply the count of the 'maximum' number of times a given motif or pattern of interest appears in a network.

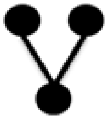
The second measure is called the F2 measure. For this measure, no two motifs can have an overlapping edge. That is, all the motifs participating in the F2 count will be edge disjoint; they won't have any edge in common. So, for a particular pattern of interest, in a given network, the F2 count is always lesser than or equal to the F1 count. The third measure F3 is both edge and node constraint but we do not address the F3 measure in our algorithm.

The basic motifs or patterns of interest

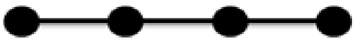
These are the basic motifs that we have focused on for this project. These are the basic building blocks using which larger motifs can be built:



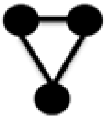
Pattern 1



Pattern 2



Pattern 3



Pattern 4

We have, at first, determined the F1 and F2 counts of these motifs in the initial network N1 (using a method for finding motif frequency in a static network). After that, the network has changed or evolved into network N2. Using the information about the edit operations which the network has gone through, we have updated the F1 and F2 counts of these motifs in the network N2.

Method for finding motif frequency in the initial network

We have used existing algorithms developed by Rasha Elhasha for the determination of motif frequency in a static network.

Method for updating the F1 and F2 counts in the changed network

Let us now consider that the target network, N1 has evolved to network N2. The change can be represented using 4 operations: Edge deletion, Edge addition, Node deletion and Node addition.

In this project, we have considered the first two operations: Edge deletion and Edge addition as Node addition and Node deletion can also be represented in terms of these operations. If a Node is deleted, it is equivalent to deleting all the Edges connected to that node. Similarly if a Node is added, we can represent that using edge addition operation.

Case 1: Edge deletion

When an edge is deleted, we have to consider that edge and find out all Matches for a certain pattern involving that edge. For this we only have to look at the neighbourhood of that edge and not search the entire network. We reduce the F1 count by as many matches as we find.

For updating the F2 count, we first see if that edge is involved in the F2 count for that pattern. If it is involved, we reduce F2 count by 1. Then we consider that Match which is present in the F2 count AND includes that edge. One by one we consider all the other edges of that Match. We try to form patterns using those edges and edges that are already not involved in the F2 count. If we get a successful match, then the F2 count is increased.

Steps

1. Form all possible patterns with that edge. Reduce the F1 count by as many matches found.
2. See if that edge is involved in the F2 count. If it is involved, reduce F2 count by 1.
3. Take the Match which is involved in the F2 count and consider each of its edges one by one.
4. For each edge form patterns involving that edge AND edges which are not there in the F2 count. If no such pattern can be found then leave F2 count as it is. If one (or more matches can be found) increase F2 count by as much.
5. Update the Edge matrix and the F2 matrix.

Case 2: Edge addition

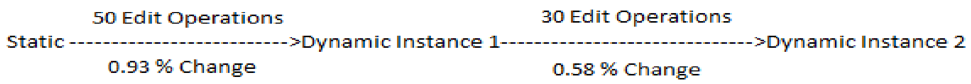
When an edge is added, we consider all the Pattern matches which can be formed using that edge and increase the F1 count by as many matches discovered. Then for each Match we check if their other edges are involved in the F2 count. If they are not involved, we increase the F2 count by 1 and include that Match in the F2 matrix.

Steps:

1. Consider all pattern matches which can be formed using that edge. Increase F1 count by as many.
2. For each such Match check state of the edges. If for any match, all edges has state 0, increase F2 count by one. Include that edge in the F2 matrix. Update the Edge matrix accordingly.
3. Update Edge matrix and F2 matrix.

Implementation and results:

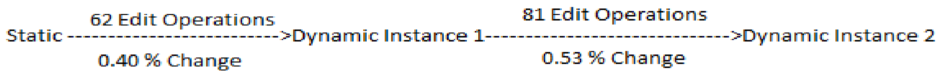
For this project, we have implemented only for Patterns 1 and 3. While implementing our method on small networks, we have got satisfactory and expected results. For larger networks, we have achieved results that are close to the optimal result in a very short time.



Dataset1: 6239

RREB1 ras responsive element binding protein 1 [*Homo sapiens* (human)]

	No. of Nodes	No. of Edges	Pattern 1	Pattern 3	Running Time(in sec)
Static	3248	5358	f1= 433 f2 =133	f1=2815598 f2=1626	12.93
Dynamic Instance 1	3211	5280	f1= 431 f2 =133	f1=2717518 f2=1593	0.275
Dynamic Instance 2	3222	5304	f1= 435 f2 =135	f1=2717672 f2=1603	0.146



Dataset2: 9606

Homo sapiens (human)

	No. of Nodes	No. of Edges	Pattern 1	Pattern 3	Running Time(in sec)
Static	6141	15513	f1= 4033 f2 =761	f1=19533653 f2=4650	135.03
Dynamic Instance 1	6139	15483	f1= 4019 f2 =749	f1=19411542 f2=4632	3.41
Dynamic Instance 2	6125	15454	f1= 3962 f2 =741	f1=19189890 f2=4621	3.58

Dataset3: 210

MIR210 microRNA 210 [*Homo sapiens* (human)]

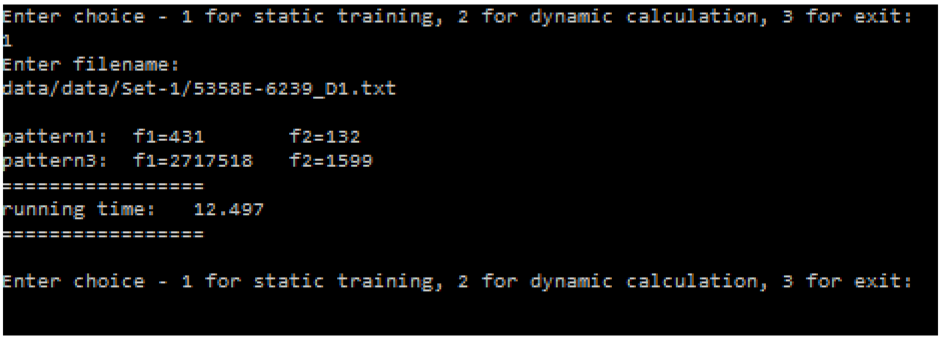
	No. of Nodes	No. of Edges	Pattern 1	Pattern 3	Running Time(in sec)
Static	733	1507	f1= 92 f2= 49	f1= 131511 f2= 472	0.829
Dynamic Instance 1	729	1495	f1= 92 f2= 49	f1= 128549 f2= 468	0.031
Dynamic Instance 2	726	1481	f1= 91 f2= 49	f1=128013 f2=461	0.016

Error Percentage Calculation

As we can see from the table, by updating the counts, we can get reasonably good predictions for the dynamic network in a very short time without doing the computation from scratch. Our result is not the optimal result and as the network changes by a large degree, the error in our estimate increases.

Sample output for dataset1 6239:

Running Static Algorithm on the Dynamic Instance 1 gives us the output as:

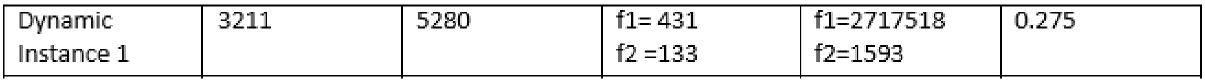


```
Enter choice - 1 for static training, 2 for dynamic calculation, 3 for exit:
1
Enter filename:
data/data/Set-1/5358E-6239_D1.txt

pattern1:  f1=431       f2=132
pattern3:  f1=2717518   f2=1599
=================
running time:   12.497
=================

Enter choice - 1 for static training, 2 for dynamic calculation, 3 for exit:
```

From our dynamic update algorithm we get the output as from the table seen before:



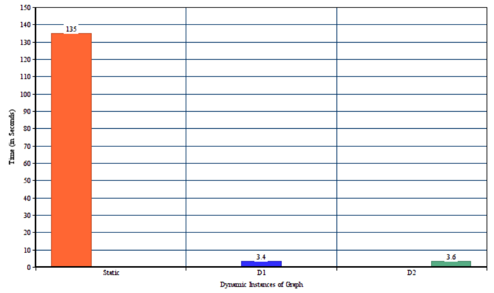
Dynamic Instance 1	3211	5280	f1= 431 f2 =133	f1=2717518 f2=1593	0.275

Error Percentage of f2 measure for pattern3 = (1599-1593)/1599 x 100 = 0.375

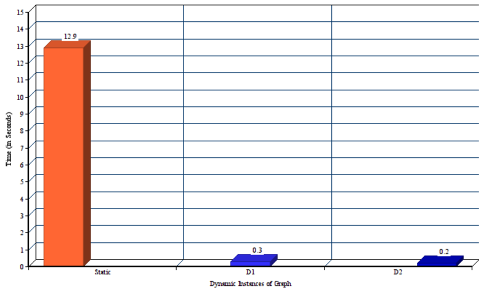
Error Percentage of f2 measure for pattern1 = (132-133)/132 x 100 = 0.75

Ratio of time taken between the static running and dynamic updation

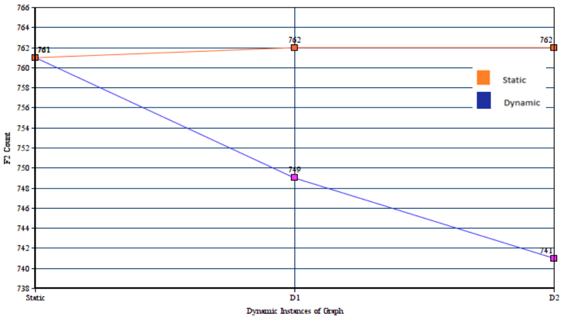
= 0.275/12.497 =0.022



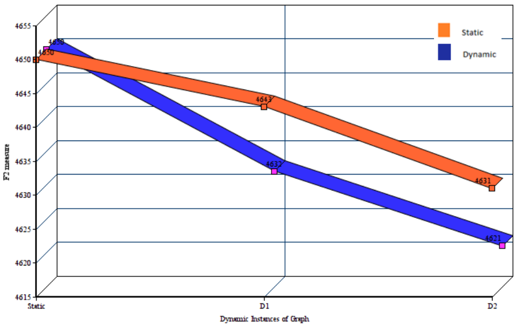
Time taken for F1 and F2 computation in static and dynamic instances of Dataset 2 (9606)



Time taken for F1 and F2 computation in static and dynamic instances of Dataset 1(6239)



F2 Computational accuracy for Pattern 1 in Dataset 2 (9606)



F2 Computational accuracy for Pattern 3 in Dataset 2(9606)

We plan to improve our implementation by using a better data structure and improving our algorithm to handle even larger networks. We also plan to work on the other two patterns that we did not address this time.

The greedy approach gives us sub-optimal results for the F2 count. We can get a better estimation of Motif frequencies in a short time by incorporating more cases in future.

Division of work:

Method: Kingshuk, Bijoykrishna, Pramit

Implementation: Bijoykrishna, Pramit

Report and Presentation: Pramit, Kingshuk, Bijoykrishna

References:

[1]. Falk Schreiber, Henning Schwöbbermeyer. Frequency Concepts and Pattern Detection for the Analysis of Motifs in Networks. Transactions on Computational Systems Biology III Volume 3737 of the series Lecture Notes in Computer Science pp 89-104

Datasets used:

Datasets from NCBI databank were used. The following datasets were used 9606, 6239 and 210.