Searching for Localization Motifs in 3'UTRs using Biological Networks

Qasim Ahmed
University of Ottawa's School of Information Technology and Engineering

Mathieu Lavallée-Adam
University of Ottawa's Department Biochemistry,
Microbiology and Immunology

Marcel Turcotte
University of Ottawa's School of Information Technology
and Engineering

Localization motifs are a very commonly used mechanism used by most known organisms. It is a crucial mechanism for the emergence of complex life. Given its importance we have relatively little knowledge of localization motifs and this is due to the nature of the 3'UTR, where they are though to be mostly located in. Finding new localization motifs is very important if we want to increase our insights into biochemical processes that are found throughout nature. In this analysis we attempt to use computational methods to discover new localization motifs using a biological network derived from multiple BioID experiments. Using the network as a complete graph made of proteins as vertices and BioID correlation values as edges, we search for motifs that are associated with proteins that have a much higher than normal correlation values. These higher than normal values suggest that those proteins are always found together in a specific area of the cell. This should lead us to conclude that the motif that is associated to all those proteins is likely a localization motif responsible for the proteins location in the cell. After we give all the motifs scores for how likely they are to be localization motifs we determine the significance of each score and we correct for multiple hypothesis testing. What we found through our analysis is a great number motifs with high scores and low p-values, but there is very high false discovery rate in our analysis. This analysis shows great promise for the discovery of localization motifs using BioID biological networks but there more improvements need to be made in the process of generating scores such that the false discovery rates would be lower.

Introduction

Life has had countless years to change and evolve into what it is today. The complexity that has arisen over that enormous amount of time is astonishing. This level of complexity would not be possible without the use of an mRNA localization mechanism. This is especially true for specialized neurons, because the site of transcription can be very far from the final location of the protein. An increasing amount of research is suggesting that the 3'UTR is where most localization motifs can be found (Andreassi & Riccio, 2009). Identification of localization elements in 3'UTRs has been difficult and we have found relatively few as a result.

Armed with new data from multiple BioID experiments, and a lot of computational resources we tackle the problem of discovering new localization motifs in 3'UTRs. Data from the BioID experiments is in the form of a biological network. This biological network is stored as a table of correlation values representing protein-protein relative localization, i.e the rows and columns are proteins and the cells are how correlated are the detection of those proteins in the BioID experiments. Without the need of further processing we can consider this network as a complete graph, where the vertices are proteins and the edges are correlation values. We then treat all of the proteins that are associated to different motifs and treat them as a subgraph. If some of our subgraphs are significantly better clustered in our graph, it suggests a common localization motif for those proteins.Thus we look at different approaches for measuring how well clustered are the protein associated to different motifs in our graph. We then determine the significance of the different scores given to the motifs using a Monte Carlos Sampling approach.

Afterwards we follow the same approach found in LESMoN (Lavallée-Adam & et al, 2017) generate our p-values and correct for multiple hypothesis testing. If the proteins associated to a motif form a subgraph which is significantly clustered in our biological network, that motif is highly likely to be a localization motif and will be flagged.

Materials and methods

Inputs

Running this experiment means manipulating a few different resources. The analysis required these following ressources;

File 1 Containing 3'UTR sequences and their mRNA reference sequence ID's

File 2 Containing mRNA reference sequence IDs and the protein they are translated into

File 3 Containing several protein names

File 4 Containing all of the proteins from the BioID experiments and how their correlation value to every other protein.

Building the Graph

BioID data is a particularly well suited for discovering new localization motif since it is a proximity based method approach for mapping protein-protein interactions (Lambert & et al, 2015). Figure 1 gives an overview of how BioID works.The data given from the BioID experiments are contained inside of File 4. The file is stored as a table where the first two columns are proteins and the third column is the correlation value. A short excerpt of File 4 can be seen in Table 1. The file is ordered by the first column, then the second column. To build our graph we simply iterate through the file and build a $n \times n$ protein (where n is the total number of proteins). The cell i, j in our table would represent the correlation value between protein i and protein j. That is, how often they were detected together in the various BioID experiments. An excerpt of the biological network can be found at Table 2.

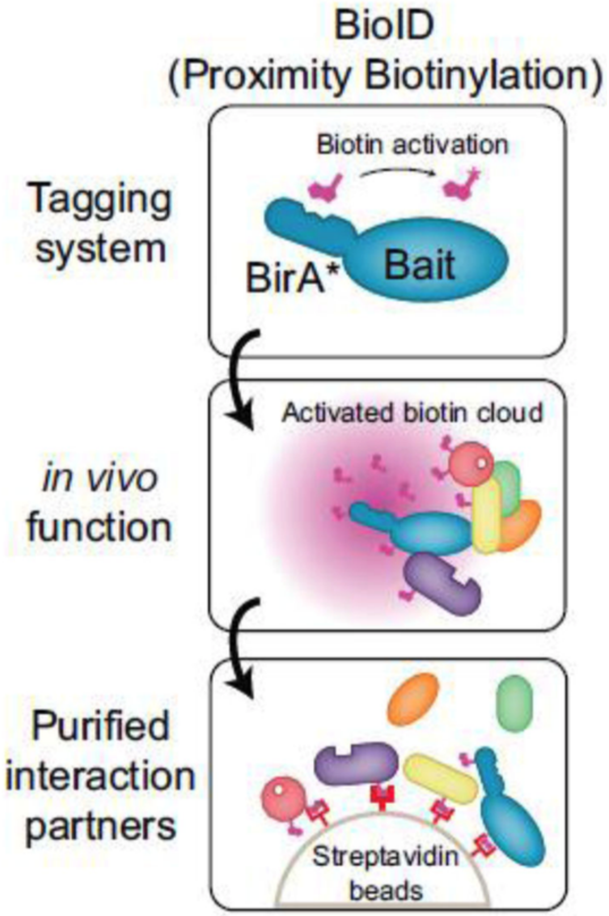
Table 1
Excerpt of File 4

ABCB7	ABCB7	1
ABCB7	DHX30	0.72388
ABCB7	SLC30A9	0.61909
...	...	...
ABCB7	DHX30	0.72388
DHX30	ABCB7	0.72388
DHX30	DHX30	1
DHX30	SLC30A9	0.81884
...	...	...

Table 2
Excerpt of Biological Network

	ABCB	DHX30	SLC30A9	...
ABCB	1	0.72388	0.61909	...
DHX30	0.72388	1	0.81884	...
SLC30A9	0.61909	0.81884	1	...
...	...	...	...	...

Figure 1. Overview of BioID



Generating Motifs

In the experiment we choose to generate all possible motifs of size 8. We also considered regular expressions of motifs. The addition of regular expressions allows us to detect different variations of a potential localization motif. To do so we included, in the standard alphabet of mRNA nucleotides, r (representing [ag], or purines) and y (representing [ct] or pyrimidines) as well as the * wildcard (representing any nucleotide or a gap). We are therefore generating all possible 8 character sequences from the following alphabet $\{a, c, t, g, r, y, *\}$. This gives in total 7^8 motifs. We then proceed with the scoring of the generated motifs.

Generating Subgraphs

With the motifs generated we would like to give each one a score —the higher the score the more likely it is to be a localization motif. But before we could score the motifs, we need to generate the associated sub graphs. It is these sub graphs that we will be scoring.

For each motif m_i,we find the mRNAs in which the

3'UTRs contain m_i, using File 1. We then retrieve the set of proteins P_i that are generated by the mRNAs, this is done using File 2 and File 3. The proteins in P_i form a subgraph in our main graph. From this point onwards whenever we refer to scoring a motif, what we really mean is scoring the subgraph associated to that motif.

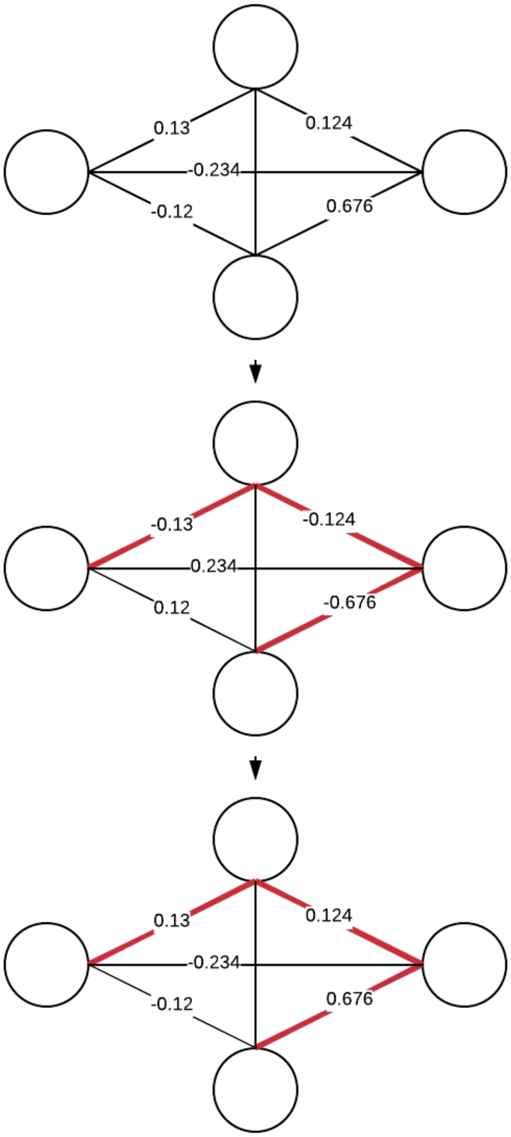
Scoring the motifs

We now have a graph represented by the proteins and their BioID correlation values, and for each motif we will have a subgraph representing the proteins associated to that motif and their correlation values. The goal is to score each subgraph so that high scoring subgraphs indicate a possible localization motif compared to lower scoring subgraphs. A good scoring measure in this case will result in a large discrepancy between the scores of localization motifs and any other motifs. In this experiment we look at two different algorithms to score our subgraphs.

Average. In this algorithm we simply take the average of all the edges in our subgraphs. Note that our subgraphs are complete, so if there are any outlier proteins they will have many associated edges hence many opportunities to contribute to the score of the subgraph.

Maximum Spanning Tree. To reduce the impact caused by the multiple outlier edges being considered for each outlier vertex in our subgraphs we try taking only the average of the edges in the maximum spanning tree of the subgraphs. To get a maximum spanning tree we first flip the signs of all the correlation values in our subgraph. We then run Prim's algorithm to find the minimum spanning tree of our negated subgraph (Minimum Spanning Tree, n.d.). We then flip the signs on the returned minimum spanning tree and what we are left with is the maximum spanning tree of our original subgraph. An example of getting the maximum spanning tree score can be seen in Figure 2.

Figure 2. Calculating MST score of 0.31



Determining Significance

Going back to the objective of finding localization motifs, once all generated motifs are given a score we would like to be able to tell which motifs are likely to be localization motifs. A score alone is not enough to arrive to any conclusions. We also need knowledge of what is a good enough score for a motif to be flagged.

Let P_i denote the set of proteins associated with motif m_i, and P_k denote a set of randomly selected proteins (without replacement) from our graph such that $|P_i| = |P_k|$ We assume the following null hypothesis;

$$H_0 : Prob[Score(P_i) > Score(P_k)] < 1 - \alpha$$

Calculating P-Values. The p-value for a motif is defined as;

$$p - value = Prob[Score(P_k) > Score(P_i)].$$

To get the true probability for m_i would involve calculating the score for all possible subgraphs of size $|P_i|$, which is intractable. Instead we use a Monte Carlos sampling approach to approximate p-values. The approximated p-values are defined as follow;

$$p - value = \frac{\#P_k : Score(P_k) > Score(P_i)}{\#P_k}$$

.

Monte Carlos Samples. Due to the Maximum Spanning Tree algorithm being $O(n^2)$ in time complexity, we limit the size of subgraphs sampled to only 900. We also ignore subgraphs of size 1 and 2, since their average and maximum spanning three scores would be identical. Thus for each different subgraph size from 3 to 900 we will take 100,000 random samples and score them.

Correcting for Multiple Hypothesis Testing. A common p-value threshold used to reject the null hypothesis is 0.05, but that is typically used for experiments which are run under 100 times. In our case after filtering the 7^8 motifs to those who are seen in 3 to 900 different mRNAs in our data, we over 2.4 million motifs to test. Simply choosing the motifs with a smaller p-value than 0.05 will allow for far too many false positives.

We first tried to correct our p-values for multiple hypothesis testing using a the Benjamini-Yekutieli approach (Benjamini & Yekutieli, 2001), but the correction was too severe and resulted in none of the motifs being flagged as significant.

We decided to calculate the false discovery rates (FDR) for several different p-values and pick a p-value that that has a low FDR while also not being high enough to allow for as many true positives to be flagged as possible.

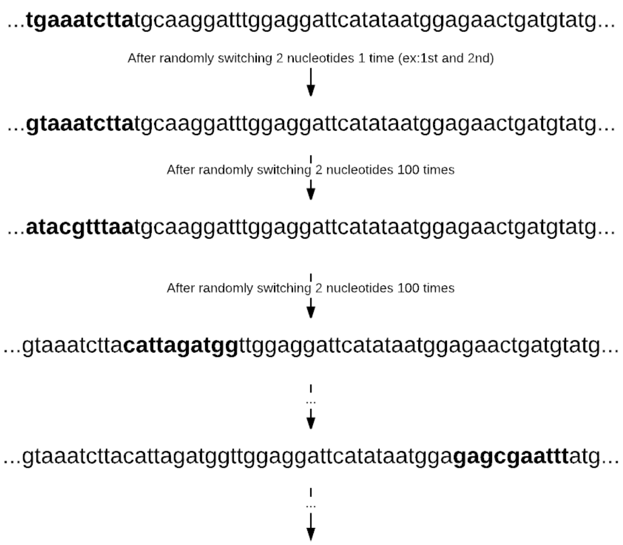
Calculating the False Discovery Ratios. It is impossible to get the true FDR for this experiment without having a complete knowledge of which motifs are false positives and that requires knowing which motifs are localization motifs. This is akin to the chicken and the egg situation. Instead we can approximate the FDR by running our discovery procedure on a dataset that is similar a property to our mRNAs as possible while not being genetic material. Because this other dataset is not genetic material it should not have ny localization motifs and as a result the associated subgraphs of the motifs built using that dataset should not have many significantly high scores.

To build this dataset we simply alter our original mRNA dataset. This is done by iterating through our dataset and for every 10 nucleotides we switch 2 randomly selected nucleotides. Two randomly selected nucleotides are switched 100 times as the sliding window goes through the mRNAs (see Figure 3). Once the entire mRNA dataset has been shuffled, we apply the same method used earlier in our original dataset. That is, with the same generated motifs, we associate each motif with a subgraph in our original graph. Each subgraph is made by matching a motif to all the shuffled 3'UTR sequences where it can be found, then associating that motif to all of the proteins produced by the mRNAs that would contain the 3'UTRs (had they not been shuffled).

For a given p-value p_i, let $M(p_i)$ and $N(p_i)$ denote the number of motifs that have a lower p-value than p_i in our original and shuffled datasets respectfully. In that case we define the FDR of p_i as follow;

$$FDR(p - value) = \frac{N(p_i)}{M(p_i)}$$

Figure 3. Shuffling mRNA nucleotides


...**tgaaatctta**tgcaaggatttggaggattcatataatggagaactgatgtatg...
After randomly switching 2 nucleotides 1 time (ex:1st and 2nd)
...**gtaaatctta**tgcaaggatttggaggattcatataatggagaactgatgtatg...
After randomly switching 2 nucleotides 100 times
...**atacgtttaa**tgcaaggatttggaggattcatataatggagaactgatgtatg...
After randomly switching 2 nucleotides 100 times
...gtaaatctta**cattagatgg**ttggaggattcatataatggagaactgatgtatg...
...gtaaatcttacattagatggttggaggattcatataatgga**gagcgaattt**atg...

Using Computing Clusters

Most of the computation being done in the entire analysis can be divided into three parts. The calculation of scores and significant values for the real dataset, real datasets and the random samples. Due to the scale of total computation needed for the three parts we had to parallelize the task. To do so we ran our work on Compute Canada's network of computing clusters. The cluster can be assigned several tasks in parallel in the form of jobs. Each job is ran on its own cpu independente from all of the other jobs.

To parallelize the analysis of the real and shuffled datasets we assigned 1 job for every possible set of motifs that can be generated by fixing the first three nucleotides. That is a total of 6^8 (1,679,616) motifs being considered for each job, with a total of 2^8 (256) job for the real dataset and another 2^8 jobs for the shuffled dataset. Submitting a job on Compute Canada is done by submitting a job script. To ensure jobs for the real and shuffled where successfully completed the sub submission scripts specified that each job required at least 16GB of ram was allocated 12h of time to run.

Parallelizing the random sampling required assigning one job to every possible subgraph size from 3 to 900. This resulted in 897 jobs being assigned where each job resulted in the generating and analysing of 100,000 random samples. To ensure jobs for the random samples were successfully completed the sub submission scripts specified also specified 16GB of ram and was allocated 24h of time to run.

Why we choose those settings for each job will be briefly explained in the discussion section.

Results

Random Samples. Figures 4 and 5 show the distribution of random samples of 500 for both the averaging and maximum spanning tree approaches respectfully. All of the other sizes from 3 to 900 produces similar distributions (500 is arbitrarily chosen to be displayed), but with the trend that larger subgraphs tend to score higher (shifting the distribution to the right).

Figure 4. Score of Averaging Approach on Samples of Subgraphs of 500

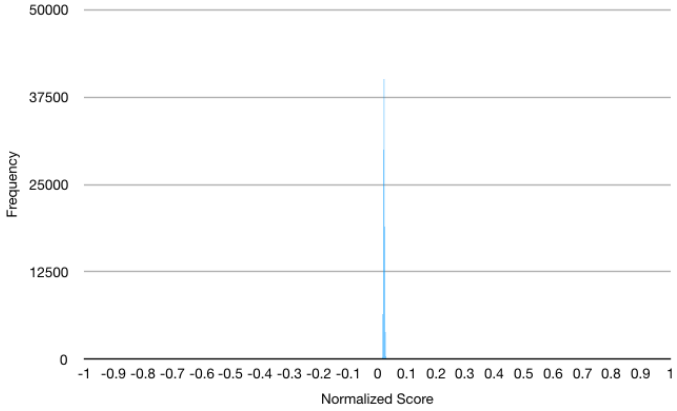
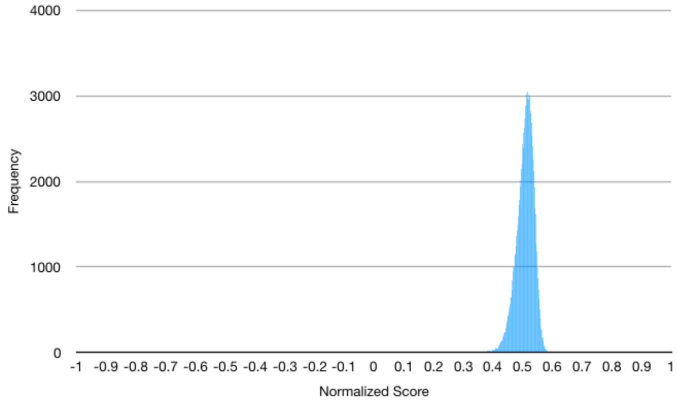


Figure 5. Score of Maximum Spanning Tree Approach on Samples of Subgraphs of 500



Real Dataset. Figures 6 and 7 show the distribution of scores of the real dataset vs the random samples for both methods. The two figures show the results for all of the motifs that form subgraphs of size 3 to 900.

Figure 6. Score Distribution of Averaging Approach on Real Dataset vs Random Samples (Logarithmic Scale)

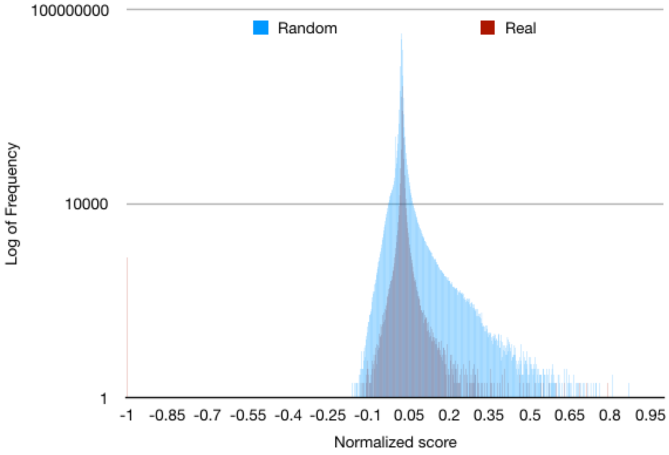
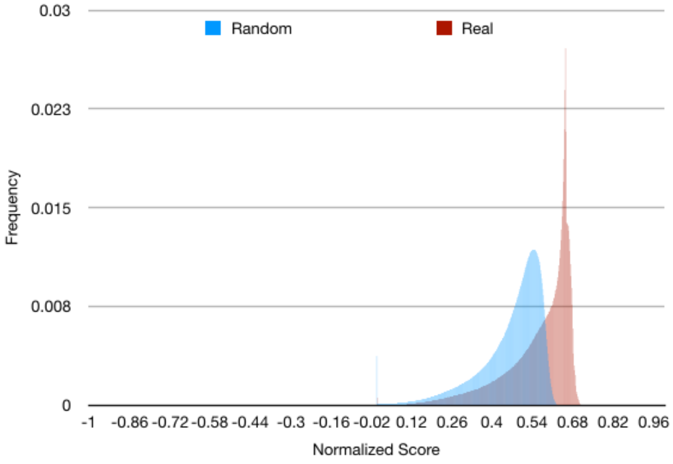


Figure 7. Score Distribution of Maximum Spanning Tree Approach on Real Dataset vs Random Samples



P-values. Using the data displayed in Figures 6 and 7 we are able to calculate the p-values for each motif. Figures 8 and 9 show the p-value results in the form of histograms.

False Discovery Rates. P-values for the shuffled datasets are also calculated and used to approximate false discovery rates as previously explained. The results of the FDR approximations are shown in Figures 10 and 11

Discussion

This experiment attempts to discover new localization motifs in 3'UTRs using a biological network. This is done by finding motifs that are associated with proteins that are very correlated in our network – significantly more correlated than random. This leads us to look for scoring measures that will give high scores to potential localization motifs and low

Figure 8. Histogram of P-Values for Averaging Approach

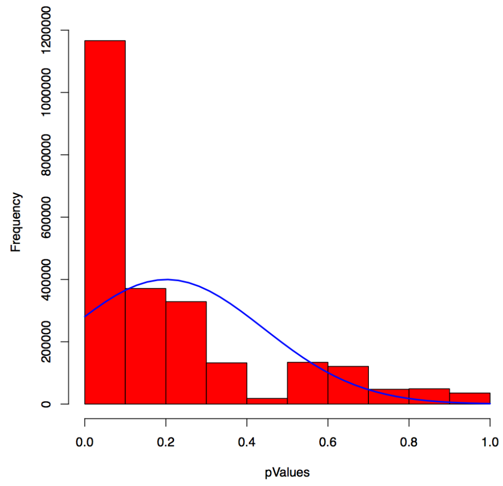


Figure 9. Histogram of P-Values for Maximum Spanning Tree Approach

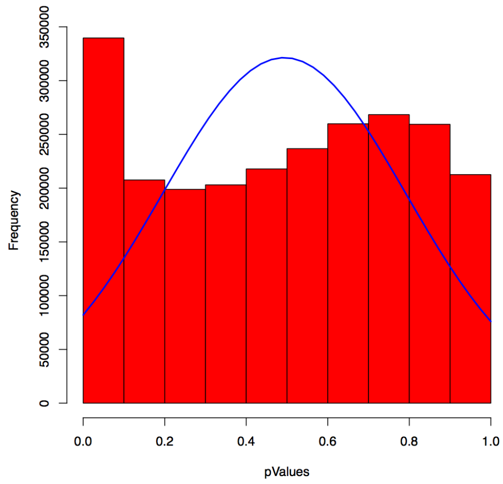


Figure 10. False Discovery Rates of the Averaging Approach

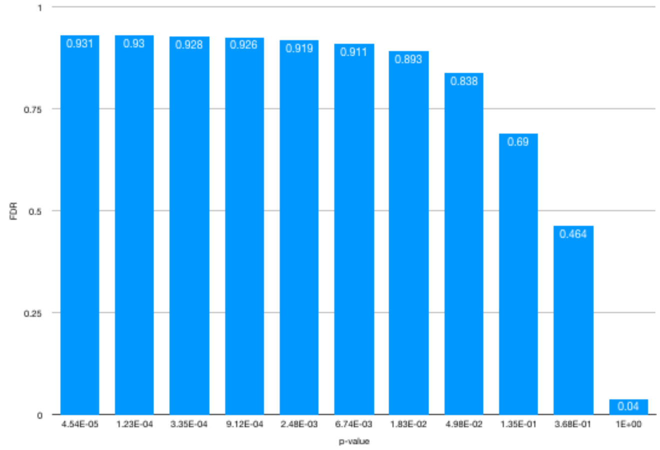
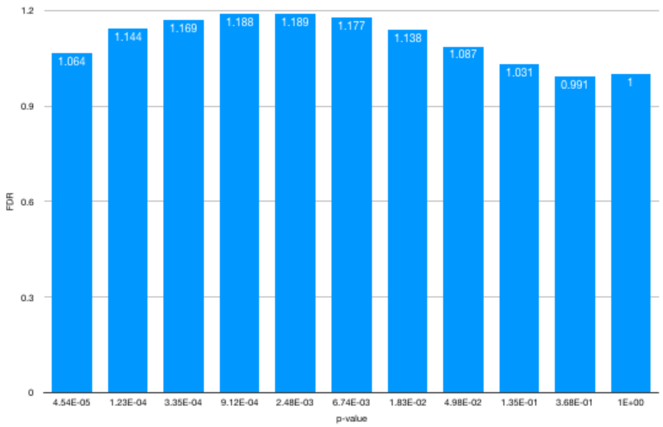


Figure 11. False Discovery Rate of the Maximum Tree Approach



scores to everything else. A good scoring measure will allow us to easily discriminate localization motifs from non-localization motifs.

Average Results. We first look at the averaging method of simply taking the average of all the edges in the subgraph associated to the motifs. We suspected this method would not do a very good job at identifying localization motifs, but we use it as a useful comparison since it is also trivial to implement. The distribution of scores seen in Figure 10 confirms our suspicious, since the distribution is very narrow. This very narrow distribution suggests this method gives most motifs a very similar scores. Figure 10 also shows why the average method is not very good. The figure shows how the distribution of the real scores overlaps with the distribution of scores in the random samples.

We suspect that the reason the average is not performing well is because the network in a complete graph. Because we have a complete graph the calculation of the average of edges is likely drowning the scores of localization motifs with a lot of noise. This is why we decided to look at the maximum spanning tree approach as we hoped it might reduce the noise.

Maximum Spanning Tree Results. Looking at figures 5 and 7 we can see that the maximum spanning tree results look promising. The distributions are a lot wider, making our analysis easier. The distribution of the ransom samples and the real dataset have significantly less overlap than the averaging method. Furthermore the real dataset distribution is shifted to the right of the random samples, which means the real dataset has significantly higher scores than the random samples.

P-values. Oddly enough when looking at the p-value histogram for the average method we see a large number of significant scores. After exploring we have found a bug somewhere in the calculation of p-values for the averaging. At the time of writing this paper we have not yet pinpointed the exact cause of the bug nor fixed it.

The maximum spanning tree also has a lot of significant p-values, and the histogram looks a lot more like is expected. That being said it is important to note cannot make any accurate claims until we correct for multiple hypothesis testing.

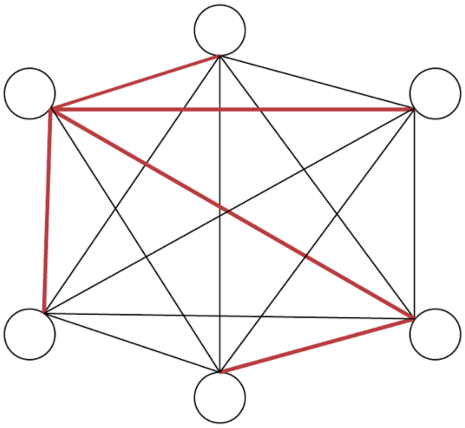
Correcting for Multiple Hypothesis Testing. Although there are many p-values with low p-values it is important to note that we expect many motifs to have low p-values by chance when we have more the 2 million tests. Without correcting for multiple hypothesis testing it we cannot make any meaningful conclusions. We cannot use a multiple hypothesis correction approach which does not take into consideration dependence because it would result it would apply to harsh a correction. We know that the tests are dependant because for example if we detect a localization motif it will likely have variants (for example if aaaaacta is a localization motif *aaaacta is also very likely to be detected as one too). When we calculated the false discovery rates for each p-value we found the rates to be too high. This suggests that about half of the motifs at most p-values thresholds will be detected as significant by chance This is far too high to derive any conclusions. Although this analysis showed some promising results, until the FDRs are considerably lower we cannot accurately discover p-values.

Possible Explanations for FDRs. As mentioned earlier, we believe that the method of taking the average of every edge in a subgraph would lead to a lot of noise. The maximum spanning tree was an attempt to lower the noise. It is possible that the maximum spanning tree method is still not reducing the noise enough. It is also possible for the maximum spanning tree to be even more sensitive to outliers in some cases. For example in a subgraph similar to Figures 5 where one protein is highly correlated with every other protein, that outlier protein will increase the score of the whole graph. The maximum spanning tree is more sensitive to this because that protein will is n^{-1} of the edges contributing to the score instead of n^{-2}. It is very possible that there are many such outlier proteins since they would represent proteins that are found in abundance in many locations in the cell.

Another possible explanation for the large FDR's is that shuffling the 3'UTR dataset is resulting in unforeseen changes in the data characteristics. An example of which is that the motifs generate larger subgraphs. Larger subgraphs also tend to have larger correlation values. This might be contributing to the higher than expected scores of the shuffled dataset which is resulting in high FDRs.

Finally because of the time limitations of this project we did not yet fix the bug found in the analysis of the averaging method. It is something we hope will be fixed once we rewrite the code in a more robust way.

Figure 12. Example of Outlier Vertex



Future Improvements. One possible improvement that would help reduce the amount of noise in the calculations of the maximum spanning tree scores is to take a fraction of the best scoring edges, in a similar way to what is done in the LESMoN paper.

Alternatively it might be worth exploring entirely different algorithms for scoring the motifs. Algorithms that would reduce the contribution of proteins that are abundant throughout the cell are of particular interest. A modified form of the Pagerank algorithm might be worth exploring. The Pagerank algorithm could be changed to consider the total contribution score of the edges that connect to a node, and the higher that score the more we would want to reduce its contribution. Once we rewrite the code we plan on making it more modular so that any appropriate scoring algorithm can be plugged into the analysis.

Acknowledgements

We would like to thank Daniela Sosa for her contribution starting the code base. We would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for their generous grants without which it would not have been possible to work on this project; Compute Canada for giving us access to the compute clusters. Finally we would also like to thank the University of Ottawa's Faculty of Engineering and Medicine.

References

Andreassi, C., & Riccio, A. (2009). To localize or not to localize: mrna fate is in 3âĂšutr ends. *Trends in Cell Biology*, *19*(9), 465-474. doi: 10.1016/j.tcb.2009.06.001

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, *29*(4), 1165âĂŞ1188. doi: 10.1214/aos/1013699998

Lambert, J.-P., & et al. (2015). Proximity biotinylation and affinity purification are complementary approaches for the interactome mapping of chromatin-associated protein complexes. *Journal of Proteomics*, *118*, 81-94. doi: 10.1016/j.jprot.2014.09.011

Lavallée-Adam, M., & et al. (2017). Functional 5' utr motif discovery with lesmon: Local enrichment of sequence motifs in biological networks. *Nucleic Acids Research*, *45*(18), 10415-10427. doi: 10.1093/nar/gkx751

Minimum Spanning Tree. (n.d.). *Primmst.java.* Retrieved from `https://algs4.cs.princeton.edu/43mst/PrimMST.java.html`