# Searching for Localization Motifs in 3'UTRs using Biological Networks

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

Mathieu LavalleÃl'-Adam
Univeristy of Ottawa' Department Biochemestry, Microbiology and Immunology

This is an example of a journal article using the apa6.cls document class to typeset manuscripts according to 6th edition of the Americal Psychological Association (APA) manual

## 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 collumns are proteins and the cells are how correlated are the detection of those proteins in the BioID experiments. Without the need of furthur processing we can consider this network as a complete graph, where the vertices are proteins and the edges are correlatation 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. Finnaly we correct our p-values 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.

Before the correction for multiple hypethesis testing the approach flagged many motifs as potential localization motifs. However we calculated a very high false discovery rate for the algorithm at many different p-value cutoffs. Therefore at the moment, we were not able to confidently say that any of the motifs flagged are localization motifs, but the non corrected p-values suggest a lot of potential in following this approach.

## Materials and methods

### Inputs

Running this experiement means manipulating a few different ressources. Past work has given us access to the 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 expriments and how their correlation value to every other protein.

### Building the Graph

The data given from the BioID experiements are contained inside of File 4. The file is stored as a table where the first two collumns are proteins and the third collumn is the correlation value. A short excerpt of File 4 can be seen in Table 1. The file is ordered by the first collumn, then the second collumn. 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 where detected together in the various BioID experiments. An exerpt of the biological network can be found at Table 2.

### 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

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 | ... |
| ... | ... | ... | ... | ... |

so we included, in the standard alphabet of mRNA nucleatides, r (reperesenting [ag], or purines) and y (representing [ct] or pyrimidines) as well as the * wilcard (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 retreive 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 discrepency between the scores of localization motifs and any other motifs. In this experiement 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 egdes hence many opportunities to contribute to the score of the subgraph.

**Maximum Spannign Tree.** To reduce the impact cause 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 1.

### 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 replecement) 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;

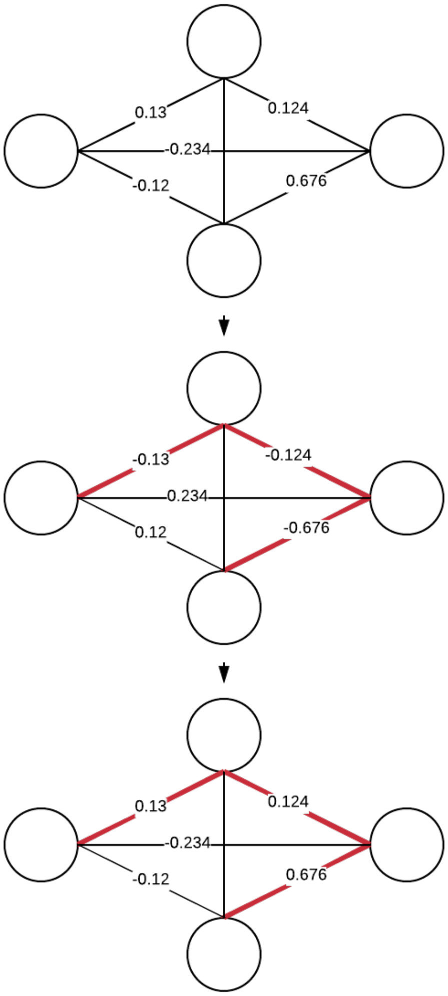
$$p - value = Prob[Score(P_k) > Score(P_i)].$$

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;

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

.

*Figure 1*. Calculating MST score of 0.31



**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 Multple Hypothesis Testing.** A common p-value threshold used to reject the null hypothesis is 0.05, but that is typically used for expriments 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 sever and resulted in none of the motifs being flagged as significant.

We decided to calculate the false discovery rate's (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 flages as possible.

**Calcuilating the False Discovery Ratios.** It is impossible to get the true FDR for this experiment without having a complete knowlege 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 discorvery procedure on a dataset that as similar a property to our mRNA's 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 assosiated 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 2). 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 mRNA's 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 orignal and shuffled datasets respectfully. In that case we define the FDR of $p_i$ as follow;

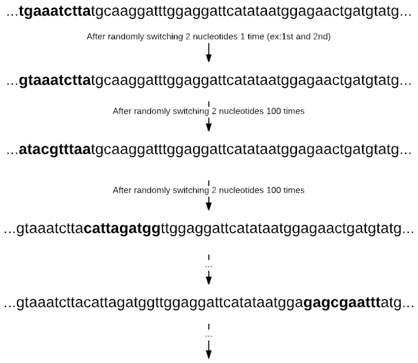
$$FDR(p - value) = \frac{N(p_i)}{M(p_i)}$$

## Using Computing Clusters

Most fo the computation being done in the entire analysis can be devided into three parts. The calculating 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 parralelize 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 parrelale in the form of jobs. Each job is ran on its own cpu independente from all of the other jobs.

To parralelize 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

*Figure 2*. 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...
...

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 16 GB of ram was allocated 12h of time to run.

Parralelizing 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 where successfully completed the sub submission scripts specified also specified 16 GB 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

## 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

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