Bioinformatics - Theory and Applications
L519 - Term Project Report

Effectiveness Of Motif Detecting Methods
A Case Study With Noisy Data

Arvind Gopu
[agopu@cs.indiana.edu]
Graduate Student
Dept. of Computer Science/Bioinformatics
Indiana University, Bloomington, IN 47405.

Wednesday, May 9th 2003

Contents

Abstract

This report explains results of experiments we conducted to study the behavior of motif discovery algorithms like MEME, etc. in noisy conditions of input data. We claim that existing motif detection algorithms do not work well when the signal-to-noise ratio (S/N) is low in the given data. This means the sequence data input to motif detecting programs should be well clustered with high S/N ratio for the algorithm (tool) to produce good results. To coroborate our claim, we tested MEME - the best known motif discovery tool - with data differing in S/N. Our results indicate that our claim is indeed a valid one that it is worth while looking into ways of improving the effectiveness of motif discovery tools.

In our experiment we selected 3 random COG families and built motifs on them considered one at a time initially, then two (merged together) and finally three (merged together) at a time. Note that the S/N ratio will gradually decrease in this kind of a setup. We noticed a clear drop in the performance of MEME in terms of motifs discovered in the 3 different cases. We have explained our results in the following sections. We have used entropy plots to explain this phenomenon. We claim that clustering data before using motif detection tools will result in better motif discovery. We have nothing in this report to support that claim though, that is something we plan to work on in future.

1 Introduction

The purpose of this report is to explain the results of our experiments on Motif detecting tools like MEME, etc. using data which differed in its signal-noise ratio. Our expectation was that the existing motif detecting tools will be less effective in a more noisy environment. So unless we have highly clustered families of proteins their usefulness gets lowered. Our results show that our expectation was indeed valid. We also have tried to give some reasons for this behavior - some statistical explanation and entropy plots.

We chose MEME for our experiments since it is one of the most well known motif discovery tools. It uses a statistics based algorithm which builds motifs according to probabilistic models of the given data. The algorithm is explained in Bailey [1] and we could say that it implicitly requires a good signal-noise ratio. The various steps followed in our project are explained in Section [2]. Section [3] gives a tabular presentation of MEME results in terms of number of motifs detected. Section [4] explains with the help of Relative Entropy plots, the effect of noise on the performance of MEME. Finally we conclude by suggesting further work related to this project in Section [5]. Information about various scripts used and the result (output) files is given in the Appendix.

2 Experiment

2.1 Motif Building

In our experimental setup, we picked 3 random COG families - COG0001, COG1959 and COG3121. We built motifs using MEME for the 3 families using the following steps:

- We ran MEME on the COGs taken one at a time. We set the maximum number of motifs (parameter $-nmotifs$ of the MEME program) at 5.
- Next, we merged the COGs two at a time and built motifs again. We set the maximum number of motifs (parameter $-nmotifs$ of the MEME program) at 10 this time around.
- Finally we merged all three families into one and used MEME to build motifs with the maximum number of motifs set to 15.

We used different number of maximum motifs to give the algorithm a fair chance, beacause there is no way we can judge its performance in detecting motifs from 2 COGs (mixed up) if we use the same number of motifs as we

used for an individual COG! Of course, keeping in mind the way MEME works, it is not that straight forward - MEME builds motifs in no particular order. But still, increasing the number of motifs is a necessary step.

We used parallel MEME to do all the computations. The 'memejob' script written by Dick Repasky does a great job, making life simple when one wants to submit parallel MEME jobs. More information about parallel job submissions on IBM SP (which we used for this project) is available in [2].

Information about the result or output files of the above mentioned steps is given in the Appendix.

2.2 Entropy Information Generation

After building motifs, we shifted our focus to the information content in the various data used. To compute entropy information, the following steps were done:

- We built multiple sequence alignments of all the COGs mentioned in the previous section.
- Then we used a script to do some format conversion on the alignment file. This step was required because the script which we used to generate relative entropy plotting data required its input to be in a particular format - sequence id, followed by a tab stop and then the multiple sequence alignment in a *single* line.
- Next, we used a script[1] to generate plotting data which we fed into Matlab. This step generates a file which can be executed directly inside Matlab without much manual intervention.

Finally we used Matlab to generate plots and saved it to be used in our report. Information about the plot-data and various scripts used is given in the Appendix.

[1]Thanks to Zhiping Wang (zhipwang@indiana.edu) who wrote this script. We modified it a bit to generate plot data.

3 MEME Results

The results of our experiments in building motifs (See section [2]) are given below. We have 3 subsections each corresponding to the 3 steps mentioned in Section [2.1].

3.1 COGs taken ONE at a time

The following table shows results of MEME for the 3 COG families taken one at a time.

COG family	Number of Motifs
COG0001	5
COG1959	5
COG1959	5

3.2 COGs taken TWO at a time

The following table shows results of MEME for the 3 COG families taken two at a time.

COG family	#	Number of Motifs		
		COG0001	COG1959	COG3121
COG0001 and 1959	10	9 (1 2 3 4 6 7 8 9 10)	1[a] (5)	-
COG0001 and 3121	10	9 (1 2 3 4 5 6 7 9 10)	-	1[b] (8)
COG1959 and 3121	10	-	3 (1 7 8)	7 (2 3 4 5 6 9 10)

[a]Motifs 7 8 occur in this - overlap
[b]Motifs 1 2 3 5 occur in this - overlap

Note that the upper value in each row (without parenthesis) indicates the number of motifs found in that COG while the values in parenthesis indicate the motif number per se. We have also included the latter because it gives a

rough idea on whether the algorithm had ample opportunity to detect motifs in sequences from all the COGs.

3.3 COGs taken THREE at a time

The following table shows results of MEME for all the 3 COG families merged together.

COG family	#	Number of Motifs		
		COG0001	COG1959	COG3121
COG 0001,1959 & 3121	15	9 (1 2 3 4 6 7 8 10 15)	1 (5)	5[a] (9 11 12 13 14)

[a]Motif 1 occurs in this - overlap

Again, note that the upper value in each row (without parenthesis) indicates the number of motifs found in that COG while the values in parenthesis indicate the motif number per se. We have also included the latter because it gives a rough idea on whether the algorithm had ample opportunity to detect motifs in sequences from all the COGs.

As mentioned before, further information about the result or output files that MEME produced, is given in the Appendix.

4 Relative Entropy Analysis

In this section, we have presented details about the analysis we did using relative entropy values generated in Section [2.2]. Initially we have shown the information content in the individual COG families. In the latter half of the plots, we have shown relative entropy of various merged COG families and how it is affected due to increased noise (which results from merging two or more COGs). We have also given useful information about position of motifs in the MEME output. We did this by looking at the output file by eye. This could be automated if more work is put into it. Also note that all graphs have sequence position (column number) on the x-axis and relative entropy on the y-axis.

4.1 Fairly Noiseless data - Individual COGs

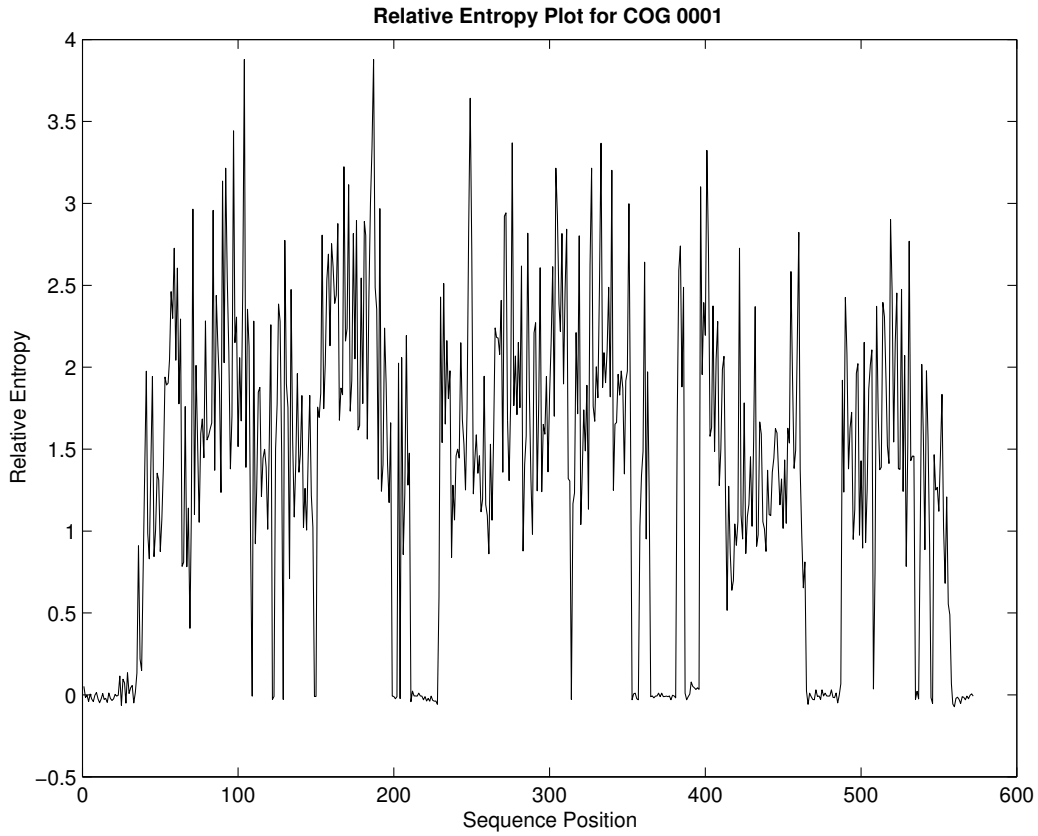

Figure 1: Entropy plot for COG0001

Figure 1 shows the relative entropy levels in COG 0001. We found that

five motifs were detected at points centered around sequence positions 50, 115, 225, 260 and 300. From Figure 1 it is clear that peaks occur around similar sequence positions on the x-axis.

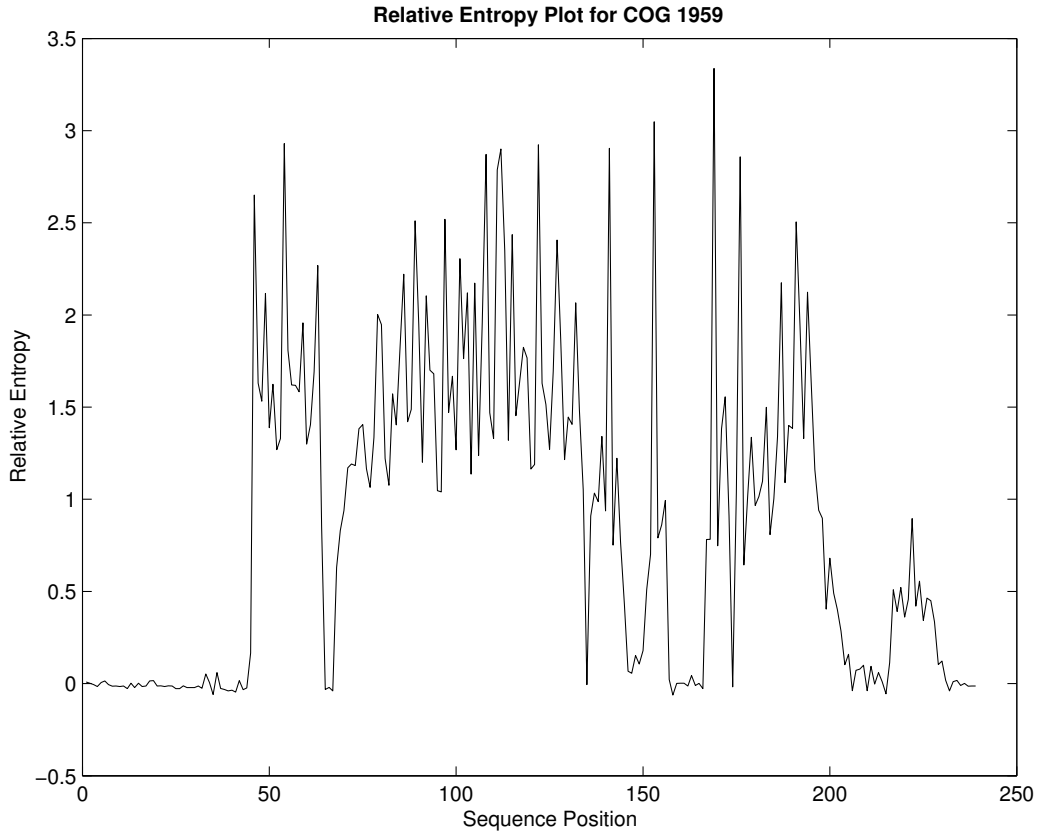

Figure 2: Entropy plot for COG1959

Similarly Figures 2 and 3 show the relative entropy levels in COG 1959 and 3121 respectively. We found that five motifs in COG 1959 were detected at points centered around sequence positions 10, 50 90, 100 and 115. The five motifs in COG 3121 were detected at points centeredd around sequence positions 40, 80, 100, 150 and 190. From Figures 2 and 3 it is clear that peaks occur around similar sequence positions on the x-axis of the corresponding plots.

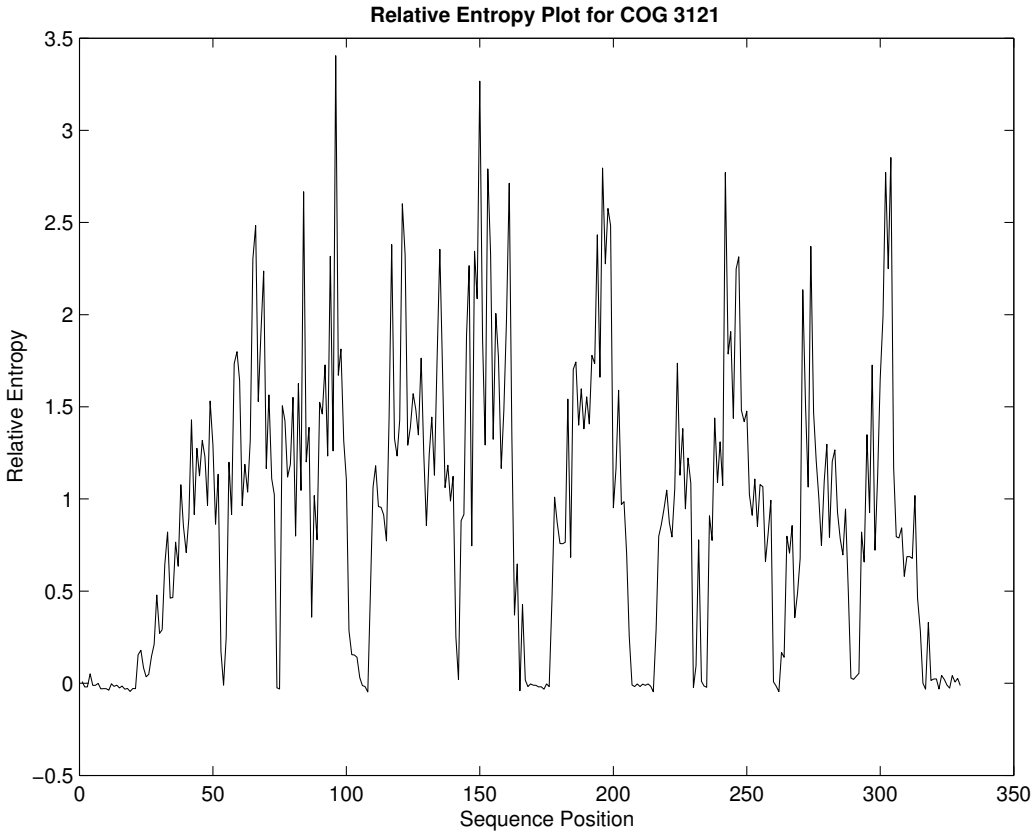

Figure 3: Entropy plot for COG3121

4.2 Noisy Data

In the following sub section, we have shown plots of relative entropy when COGs were merged 2 or more at a time. From the graphs and the accompanying explanations we try to make clear how information content and newly introduced noise messes up the motif detection i.e. Lower (S/N) ratio leads to a different set of motifs being detected with disparity in the number of motifs found, between the COGs.

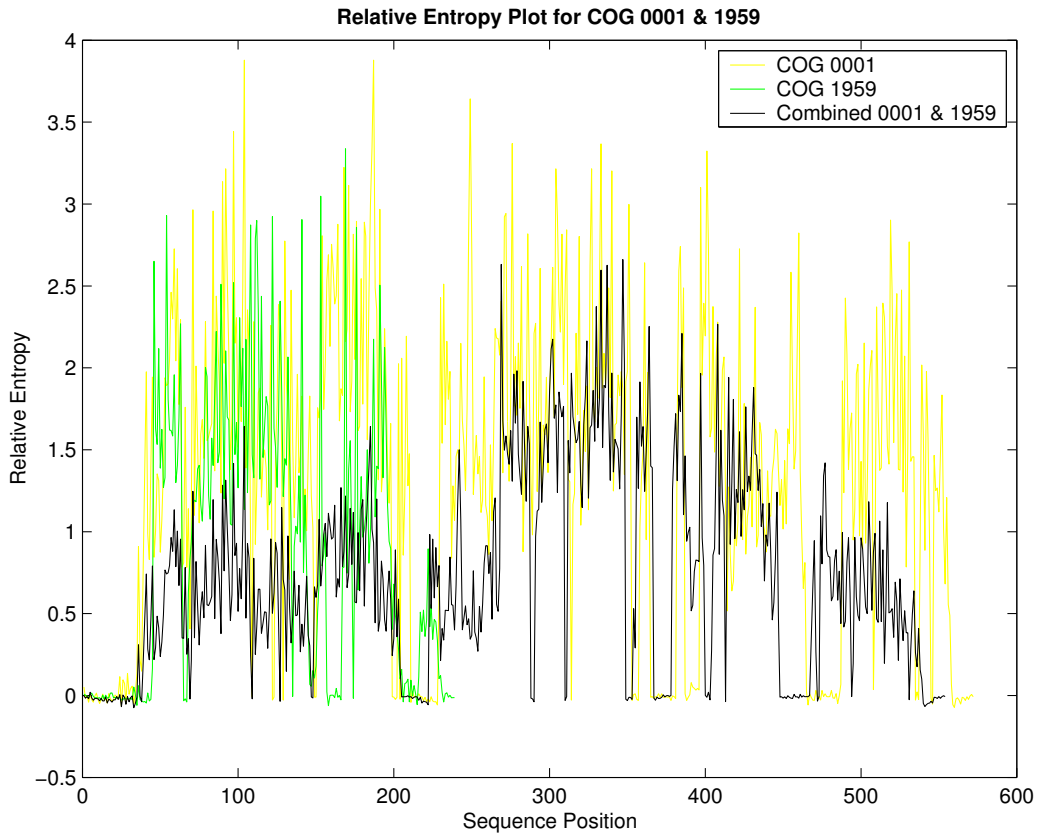

Figure 4: Entropy plot for COG 0001 & 1959 merged

Figure 4 shows the relative entropy value for sequence points when COG 0001 and 1959 have been merged. We have also shown the relative entropies of individual COGs in lighter colors. It is quite clear that the net relative entropy is lesser in areas where sequences are present from both COGs - for example at around sequence position 80, postion 125, etc. In areas where only COG 0001 has sequences, the relative entropy spikes are not affected that badly - for example at sequence position 320, etc.

The motifs detected for the merged COG showed a remarkable bias towards COG 0001 with 9 of the 10 motifs found, belonging to it. This is apparently due to the (S/N) ratio getting messed up in areas where sequences from both the families are present depending on sequence similarity and so forth. The only motif detected corresponding to COG 1959 was at around position where the relative entropy for COG 1959 is more than COG 0001.

There is another thing to note in Figure 4. A couple of motifs which are originally from COG 0001 have been detected in the portion of the alignment where COG 1959 exists. This could be due to sequence similarity between the two COGs in those areas.

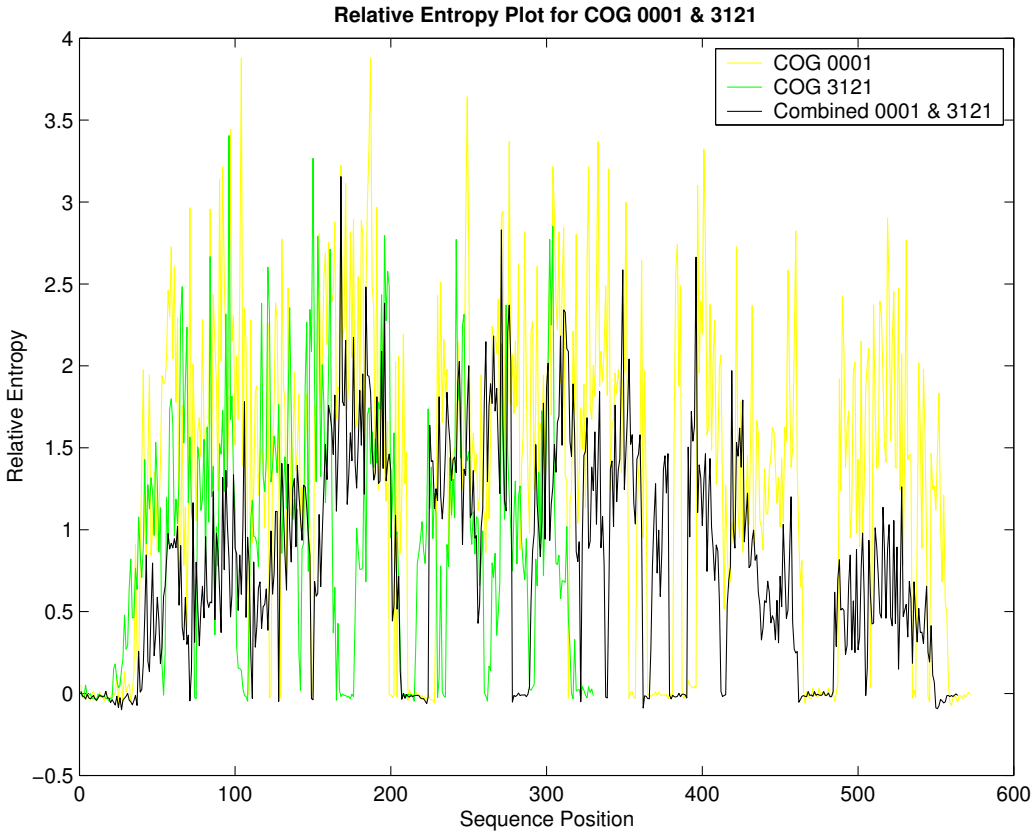

Figure 5: Entropy plot for COG 0001 & 3121 merged

Figures 5 and 6 show the relative entropy plots for the merger of COG 0001 & 3121 and COG 1959 & 3121 respectively. In the former case the detected motifs are highly biased towards COG 0001 with only 1 of the 10 motifs belonging to sequences from COG 3121 - a clear indication of the effect of noise on the performance. The only motif detected corresponding to

COG 3121 was at around position 100 where it has its highest information content (notice green spike). Again, a couple of motifs which are originally from COG 0001 have been detected in the portion of the alignment where COG 3121 exists. This could be due to sequence similarity between the two COGs in those areas.

In the second case the performance of MEME is similar to the cases mentioned for the other two 2-COG mergers. It's probably slightly better in a sense that 3 out of the 5 motifs detected in individual COG 1959 have been detected in this case too. But still COG 3121 is more dominant over 1959.

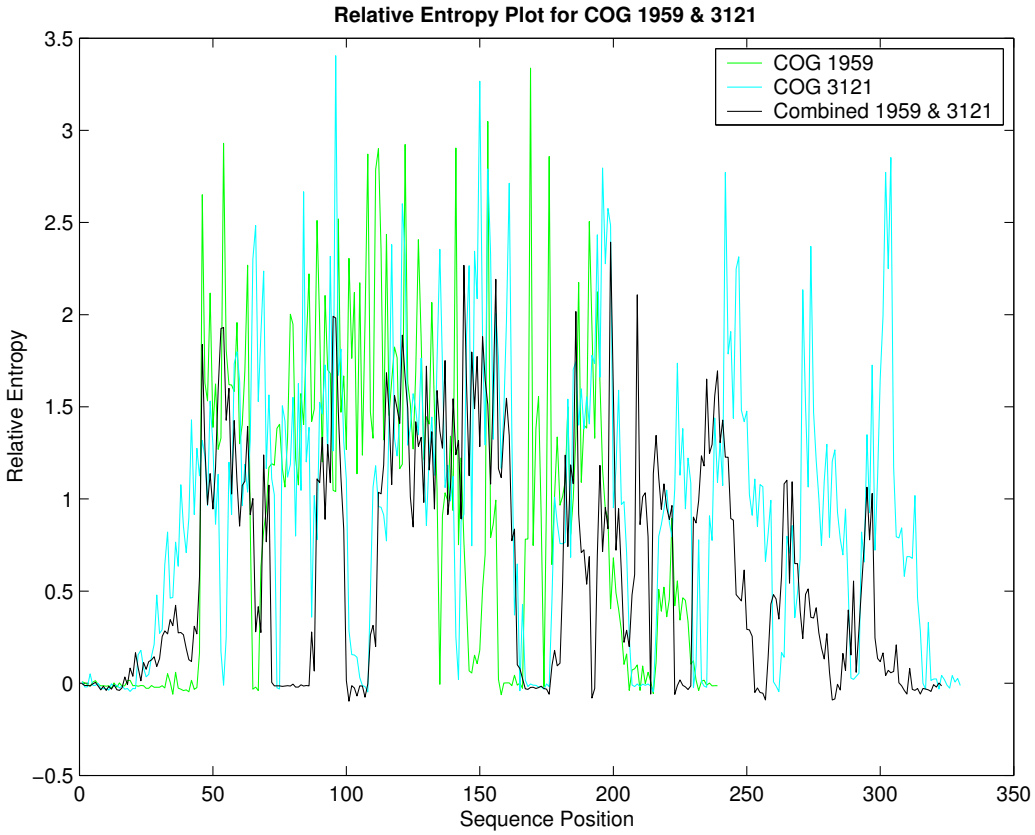

Figure 6: Entropy plot for COG 1959 & 3121 merged

4.3 Most Noisy Data

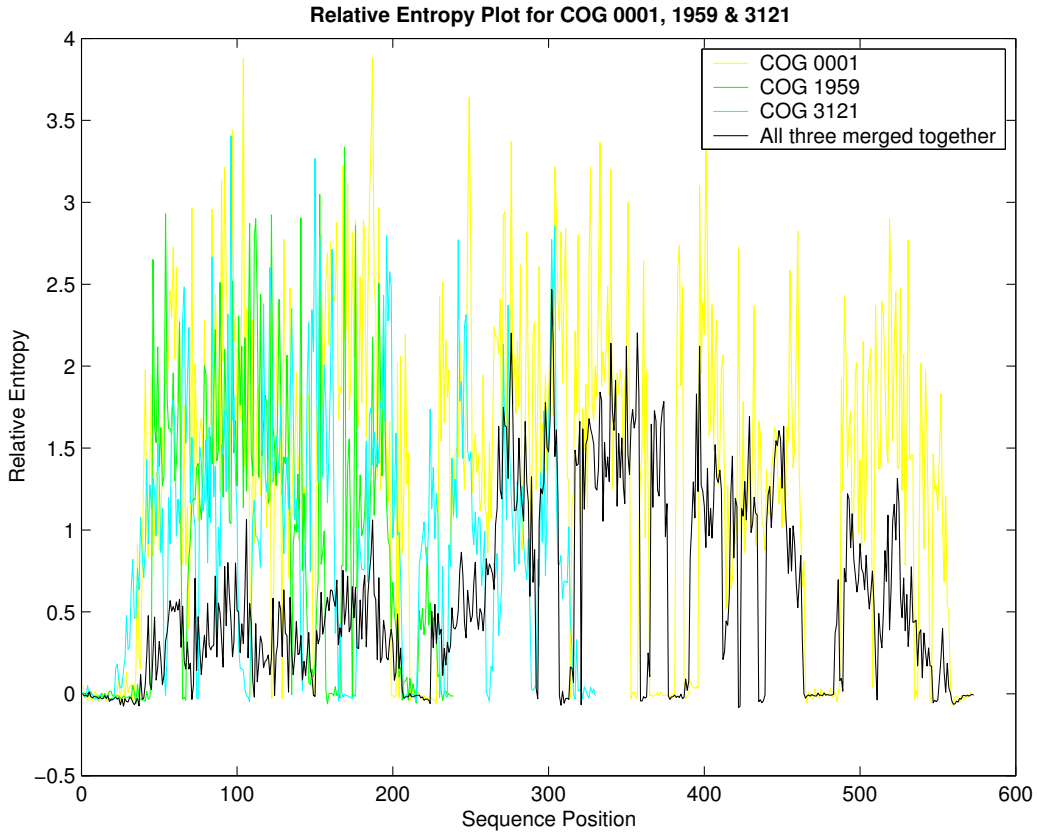

Figure 7: Entropy plot for COG 0001, 1959 & 3121 merged

Figure 7 shows the relative entropy value for sequence points when all three COG families have been merged. We have also shown the relative entropies of individual COGs in lighter colors. It is evident that the net relative entropy is highly suppressed in areas where all three sequences are present - for example at around sequence position 100. In areas where only COG 0001 has sequences, the relative entropy are not that badly affected - for example at sequence position 400.

In the case of all COGs merged together, we found that motifs were detected in sequences corresponding to COG 0001 almost through out the length of the sequences. In the case of COG 1959 only 1 motif was detected corresponding to the green spike at around position 40 - 50. In the case of COG 3121 the motifs detected were remarkably similar to the ones detected when only COG 3121 was used. This indicates that one of the COG families we selected (COG 3121) has not been affected as much as another (COG 1959 - was affected badly). There are a few factors which we can attirbute this - sequence similarity/dissimilarity between the COGs which are being merged and how the (S/N) ratio is affected as a consequence.

If you are interested in having a first hand look at the MEME output files and how the motifs correspond to the information content shown in the plots of this section, we recommend you look at the information given about our result/output files in the Appendix.

5 Conclusion

We conclude our claim that existing motif detection algorithms do not work well when the signal-to-noise ratio (S/N) is low in the given data is surely valid. Our results as explained in Sections [3] and [4] support our claim. Only if sequence data input to motif detecting programs is well clustered with high S/N ratio do these algorithms (tools) produce good results. We believe it is worth while looking into ways of improving the effectiveness of motif discovery tools. We also propose that clustering data before using motif detection tools will result in better motif discovery. That is something we will be working on in the near future.

Also we plan to experiment further using the same experimental setup but with different COGs and varying number of COGs (preferably increased to a higher value). Not only will this help us understand the behavior of motif based algorithms even better but also it will aid us in knowing for sure that our case study is not an exceptional case.

A Appendix

NOTE: This information mostly applies only to students and faculty members of Indiana University. You need to have an account on the IBM SP research system. Please email me at agopu@cs.indiana.edu if you have any questions.

A.1 Information about Result Files

Results of our experiments are available on IBM SP in various directories under /gpfs/agopu/term_project.

- "meme*" contains the MEME program results[2] in HTML format.
- "alns" contains the multiple sequence alignments produced as part of our analysis.
- "stats" contains among other things, scripts to generate statistical (entropy) data.
- "plot_data" contains the '.m' files which were used to generate plots we gave in section [4].
- "pratt" contains motifs we built using the 'PRATT' program. This was for purely experimental purposes and it produced similar results to the ones we had in previous sections.

A.2 Information about Script Files

Scripts used in our experiments are available on IBM SP in various directories under /gpfs/agopu/term_project.

- In the "stats" directory:
  - 'Convert_to_zhiping_dataformat.pl' is a script which converts basic multiple sequence alignment results (in .GDE format) to a format required by the program we used to generate entropy data (See next point).
  - 'Extract_req_from_all_cmp_perl.pl' is the script we used to generate relative entropy data. A substantial part of this program was written by Zhiping Wang as indicated previously.

[2] "memep" specifically contains results from parallel MEME executions, though the results are essentially the same

- Other than the above mentioned Perl scripts, there are quite a few shell scripts in the 'stats' and the 'aln' directories which help in calling the the perl scripts, moving files and so forth.

References

[1] Timothy L. Bailey, *Discovering Motifs in DNA and Protein Sequences: The Approximate Common Substring problem*, (1995).

[2] http://sp-www.iu.edu/SP.jobs.shtml - Information about parallel jobs on IBM SP.

[3] http://www.rice.edu/perl4lib/archives/2002-05/msg00002.html - Perl Help.

[4] http://www.pantz.org/scripting/perl/perlnotes.shtml - Perl Help.

[5] Leslie Lamport, *LaTeX User's Guide and Reference Manual*, (1994).

[6] http://www.geom.umn.edu/ rminer/tex/latex/ - LaTeX tutor.

[7] http://www.maths.tcd.ie/ dwilkins/LaTeXPrimer/ - LaTeX tutor.

Acknowledgement

I want to thank Prof. Sun Kim for his valuable suggestions and support for this project. I also want to thank graduate student Zhiping Wang for his entropy related script which was used as part of the graph-data generating script.