Motif-Finding in Trypanosomatids

Eithon Cadag, Kelan Wang
CSE527, Aut04 Prof. Larry Ruzzo

1. Introduction

Various methodologies have been developed and used to locate functional regulatory elements within genomic sequences. These motifs are generally short repeating strings with some amount of permutation, and occur upstream of protein-coding regions, sometimes signaling to transcriptomes the binding sites on the sequence. Locating these short strings are key to understanding the inner workings of the genome, including how and when particular genes may be expressed.

The following paper explores the use of several different motif-finding algorithms applied to two members of the Trypanosomatid genus, L. major, and T. brucei, eukaryote parasitic organisms, which share a high level of conservation and synteny amongst their respective genomes. The algorithms are used to locate regulatory elements in noncoding sequences of a subset of the two genomes (the subsets used are chromosomes within the L. major and T. brucei genomes which contain protein-coding regions that were annotated through manual curation or verified in the wet lab) The data from the three algorithms is then compiled and used to draw conclusions about the presence and specification of motifs within the Trypanosomes.

2. Trypanosome Biology

Several types of Trypanosome have shown to have a high level of conservation between speices. Both L. major and T. brucei have genes whose content are similar enough in both sequence and relative position to other genes that they are syntenous. Entire chunks of L. major chromosomes, for instance, can be found wholly contained within a subsection of a T. brucei chromosome, generally with some limited re-arrangement or reversal.

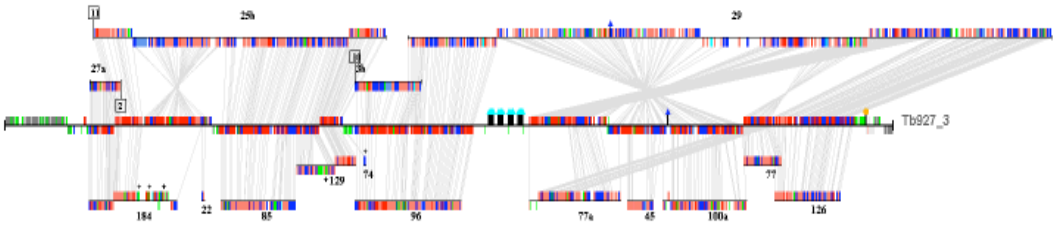
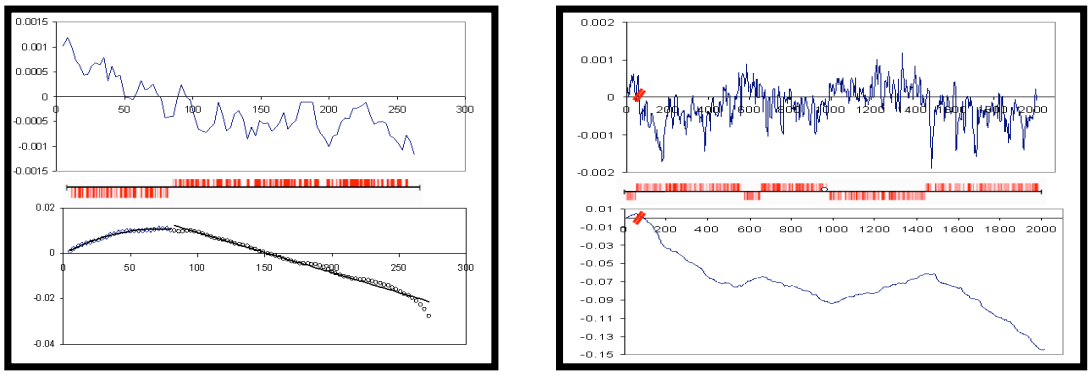


Figure 1. Synteny map of T. brucei and L. major

Additionally, the gene organization of Trypanosomes are organized into polycistronic regions; a sequence of genes code on one side of helix, and after a strand-switch region, a series of genes will then code on the other side of the helix. This effect is visible on the synteny map (Figure 1), as large tracts of genes appear on one strand, then switch to another, and then back, etc. Locations of strand-switch regions, and whether or not a section of genes will appear on the upper or lower strand, has found to be predictable through the cumulative guanine-cytosine bias (GC skew) of a sequence (Myler, et. Al 2000). Points at where the GC skew reach a valley or peak are generally strand-switch regions, and slopes are tracts of the sequence where genes appear on only one side or the other. In L. major, genes appear on the top strand if the skew is negative; the inverse is true in T. brucei.



Figures 2 and 3. The two panels are L. major chromosomes: the top frame of each is the (non-cumulative) GC skew, the middle is the strand-location of genes, and the bottom frame is the cumulative GC skew. Note that at the peak of the cumulative skew, the strand position of the genes change.

While most of the genomes of both L. major and T. brucei have not been completely officially annotated, the unique characteristics of Trypanosomatids allow for reasonably accurate predictions of likely coding regions despite a lack of wet-lab verification of protein-coding ORFs.

3. Motif-Finding Algorithms

Sequence alignment is a fundamental means of detecting biologically significant patterns in biopolymers. Patterns shared by multiple protein or nucleic acid sequences shed light on molecular structure, function and evolution in organisms. The problem can be divided into "global multiple alignment", whose aim is to align complete sequences, and "local multiple alignment", which attempts to locate and align relatively short patterns shared by otherwise dissimilar sequences (Lawrence et al. 1993).

A plethora of multiple local alignment algorithms have been developed. In this project, we applied three software packages of different alignment algorithm in finding promoters of genes in Trypanosomatid: AlignACE (Hughes et al. 2000), MITRA (Eskin et al. 2002), and GLAM (Frith et al. 2004).

3.1 Variations of Gibbs Sampling Algorithm

AlignACE and GLAM are both variation of the Gibbs sampling algorithm. As first proposed by Lawrence et al. in their paper, the Gibbs sampling is based on the statistical method of iterative sampling. The algorithm finds an optimized local alignment model for *N* sequences by locating the alignment that maximizes the ratio of the pattern probability to background probability:

$$F = \sum_{i=1}^{W} \sum_{j=1}^{4} c_{i,j} \log \frac{q_{i,j}}{p_j} \qquad \text{(equation 1)}$$

where $q_{i,j}$ is the residue frequencies for each position *I* from 1 to *W*, *p* is the background model frequencies, $c_{i,j}$ is the count of nucleic acids j in this position. And it allows the simultaneous detection and optimization of multiple patters and pattern repeats. There are two basic steps in the Gibbs sampler:

1. Predictive update step:
    a. Choose one random sequence *z* from the *N* sequences, as well as random starting positions within the various sequences.
    b. Calculate pattern probability and background probability from the current positions in all sequences excluding *z*.
2. Sampling step:
    a. Calculate probabilities of generating every possible segment of width W within sequence z according to the current pattern probability (*Q*), and the background probability (*P*).
    b. The weight $A = Q/P$ is assigned to each segment and a random one is selected for the next iteration.

For multiple pattern search, several distinct patters rather than a single one is maintained.

3.2 AlignACE

AlignACE is based on Gibbs sampling algorithm and returns a series of motifs that are over-represented in the input set. It differs from this method is the following ways:

1. Both strands of the input sequence are simultaneously considered in each step of the algorithm. Overlapping sites are not allowed even if the sites are on opposite strands.
2. Simultaneous multiple motif searching was replaced by an approach in which single motifs were found and iteratively masked. The masking is done by determining the most information-rich column in each motif, mapping that column back to the input sequences, and placing a marker at each of those positions. The sample is then reinitialized to find another motif with the stipulation that no sites that contain a masking marker may be resampled. In the case of a very strong motif, it is possible for the motif to have one of its positions masked and yet still retain enough information in its other positions for a variant of the original motif to be found.
3. MAP score is the criterion on which the final output is based. The MAP score measures the degree to which a motif is over-represented relative to the expectation of the random occurrence of such a motif in the sequence under consideration. The main drawbacks of the score is the fact that some motifs occurring ubiquitously in a genome are scored very highly, but are not likely to be relevant to the specific genes in question.

3.3 GLAM

GLAM (gapless local alignment of multiple sequences) is a modification to the Gibbs sampling alignment algorithm that allows the width of the aligned motif to be discovered automatically with an alternative Bayesian scoring scheme that incorporates a prior probability distribution over the

propensities of a position in the motif containing each of the nucleotides, A, C, G, T. The prior is selected using a Dirichlet function:

$$prior\{q_i\} = \frac{1}{Z}\prod_i q_i^{\alpha_i - 1} \qquad \text{(equation 2)}$$

where Z is a normalization constant, and different choices of pseudocounts (α), and q_i the frequencies of occurrence of a base over the set {A,C,T,G}. The final alignment scoring formula then is defined as:

$$S = \sum_{k=1}^{W} \ln \left[\frac{\frac{\Gamma(A)}{\prod_i \Gamma(\alpha_i)} \frac{\prod_i \Gamma(c_{ki} + \alpha_i)}{\Gamma(N + A)}}{\prod_i p_i^{c_{ki}}} \right] \qquad \text{(equation 3)}$$

where A is the sum of α_i, N is the number of sequences in the alignment, c_{ki} is the count of nucleotide I in the kth column of the alignment, and p_i is the background frequency model.

Beginning from a completely random alignment, the algorithm proceeds through many iterations. At each iteration, it randomly selects one sequence, and stochastically alters which segment of that sequence to include in the alignment. The probability of choosing each segment is proportional to *exp(S)*, where *S* is the alignment score that would result from including that segment (equation 3). The algorithm further incorporates two resizing moves. In the first kind of move, the left ends of the aligned segments in each sequence are held fixed, and the right ends are varied to change the width of the alignment. A new width is chosen with probability proportional to *exp(S)*. For the second move, the right ends are held fixed and the left ends are varied. As an added benefit, these resizing moves overcome a problem of the fixed width algorithm, which can get stuck in alignments whose end points are shifted left or right relative to the optimal. The search continues until a certain number of iterations have passed since finding the best alignment score so far. In addition, the program performs multiple runs of the search algorithm for a given alignment problem. If many of these runs converge to the same best alignment, then it is assured to be a global optimum.

3.4 MIsmatch TRee Algorithm (MITRA)

MITRA is a composite pattern discovery algorithm that uses a mismatch tree data structure to split the space of all possible patterns into disjoint subspaces that start with a given prefix. By splitting the pattern space, MITRA keeps reducing the pattern discovery into smaller sub-problems. In this context, the pattern discovery problem is defined as the search of all *l*-mers (a continuous string of length l) that occur with up to d mismatches in at least k sequences in the sample S.

A pattern is called weak if its has less than *k (l, d)*-neighbors (all possible *l*-mers with up to *d* mismatches as compared to the canonical pattern) in the sample. A subspace is called weak if all patterns in this subspace are weak. For example, if we are looking for patterns of length *l*, we would first split the space of all *l*-mers into 4 disjoint subspaces (A, C, G, T) and the determine whether the subspace contains a *(l,d) - k* pattern.

Mismatch tree is rooted tree where each internal node has 4 branches, each labeled with a symbol in {A, C, G, T}. The maximum depth of the tree is l. Each node in the mismatch tree corresponds to the subspaces of patterns P with a fixed prefix and contains pointers to all *l*-mers instances from the sample that are within d mismatches from a pattern p. The MITRA-Count algorithm is as follows:

1. Initialize the root node that corresponds to the set P of all 4^l *l*-mers of length *l*. This node points to all *l*-mers in sample. The first child, *A* is then examined. This child points to all of the *l*-mers in the sample that have prefix *A* (with 0 mismatches) and to all of the *l*-mers in the sample that have a different prefix.
2. The tree is explored in a depth first fashion for testing every node to see if it corresponds to a weak subspace. If yes, we backtrack. If the depth l is reached, then the node corresponds to an *(l, d) – k* pattern.
3. Backtrack in the tree, collapse the current node, and expand the next node.

MITRA also takes advantages of pairwise similarity between instances. These similarities can be used to construct a graph where each vertex is an l-mer in the sample and there's an edge if two *l*-mers are similar. Instances of a *(l, d) –k* pattern form a clique of size *k* in the graph. The edges are removed if they are not part of a clique. If a clique cannot be ruled out, the subspace of patterns considered are split by examining the child nodes. The MITRA-Graph algorithm leads to a more efficient pruning of the mismatch tree than in MITRA-Count. The algorithm works as follows:

1. Compute the set of edges at the root node by performing pairwise comparison between all *l*-mers.
2. Traverse the tree in depth first order passing on the valid edges and keeping track of the quantities mismatches.
3. At each node, eliminate edges that correspond to vertices that have degrees of less than *k-1*. If there are less than the minimum numbers of edges for a clique, backtrack on the tree.

4. Experiment and Results

AlignACE, GLAM, and MITRA were all ran on subsets of the L. major and T. brucei genomes. The most curated sequences were chosen from each (chromosome 1 respectively) and the protein-coding ORFs excised out of the sequences to give the non-coding upstream regions, which presumably would contain functional motifs. AlignACE and GLAM were run on Linux machines to generate results, while the data for MITRA was entered via a web interface. Unfortunately, the MITRA interface limited the size of the sequence submitted for processing, so while L. major was processed as a single unit, T. brucei was split into four separate sections.

Once the motifs were generated, they were parsed out of the output files and ran through a script to generate weight-matrix model (WMM) tables, for easier comparison between algorithms and genomes. Primary attention was given to the top five scoring motifs from each algorithm, and at our discretion others were added liberally. Those that were added that were not in the top five were still high-scoring motifs, and were generally chosen for their complexity, and thus, increased unlikeliness of being random repeating sequences.

4.1 Data Analysis

The results of each algorithm were compared between the genomes and each other to try to identify common patterns or trends. We found that a common motif that occurred among all algorithms and both genomes was a **CACA** and **GAGA** repeating motif. Both these types of motifs scored high (in the

thousands for AlignACE and GLAM), and could possibly be functional motifs. **CACA** could possibly be altered poly-A regions, or a transposable element that isn't an actual functional motif.

The individual results of each algorithm were generally very distinct. AlignACE results mostly consisted of **CACA** and **GAGA** repeats for L. major, but for T. brucei yielded relatively complex (compared to **CACA** repeats) sequences that were highly conserved between non-coding segments. As the WMM below points out, some T. brucei motifs followed a particular pattern very consistently (evident by the presence of numerous $P = 1.0$ for base probabilities).

MOTIF #1:

	A	C	T	G	N		
1	1.00	0.00	0.00	0.00	0.00	0.00	A
2	0.00	0.00	0.00	1.00	0.00	0.00	G
3	0.00	0.00	1.00	0.00	0.00	0.00	T
4	1.00	0.00	0.00	0.00	0.00	0.00	A
5	0.00	1.00	0.00	0.00	0.00	0.00	C
6	1.00	0.00	0.00	0.00	0.00	0.00	A
7	0.45	0.48	0.00	0.07	0.00	0.00	C/A
8	0.00	0.00	0.00	1.00	0.00	0.00	G
9	0.00	1.00	0.00	0.00	0.00	0.00	C
10	0.44	0.00	0.05	0.51	0.00	0.00	G/A
11	0.51	0.04	0.01	0.44	0.00	0.00	A/G
12	0.01	0.00	0.48	0.51	0.00	0.00	G/T
13	0.48	0.00	0.51	0.00	0.00	0.00	T/A
14	0.51	0.01	0.48	0.00	0.00	0.00	G/A
15	0.48	0.51	0.00	0.00	0.00	0.00	C/A
16	0.51	0.40	0.00	0.09	0.00	0.00	A/C
17	0.00	0.52	0.48	0.00	0.00	0.00	T/G
18	0.06	0.00	0.00	0.93	0.00	0.00	G
19	0.00	0.99	0.00	0.00	0.00	0.00	C

Figure 4. WMM for an **AGTA…** T. brucei AlignACE motif

GLAM was characterized with **CACA** and **GAGA** repeats in L. major and long motifs of repeating adenines, with other bases inserted at points in T. brucei. In fact, GLAM did not find any common types of motifs between L. major and T. brucei. Below is comparison of common L. major and common T. brucei motifs found by GLAM.

AGAGAGAGAGAGAGAGAG
GGAGAGAAAGACAAGCGGAG
GCAGAGAGAGAGAGAGAC
AGAACGAGAGACGGACACAG
ACGACGAGAGAGAGAGAA
AGAAAGAGAGACGGAGAGAG
GCGGAGAAAGGGAAAAGGAG
GCAGAGCGAGAGAGCGAGAG
ACAGAGAGAGAGAGAGAG
ACACAGAGAGAGAGAGAG
ACAGCGCGAGACAGAGGGAG
AGAAAGAGAGAGAAGAGGCA
AGAGAGAGAGAGAGAGAG
AGAGAGAGAGAGAGCGGGGG
AGAAGAAGAAGAAGGGGGAG
AGAAAGAGAGAGAGAGAG
AGGAAGAGAGAAAGAGAGAA
AGAGAGAGAGAGAGAGAG
AGAGAGAGAGAGAGAGAG

AAAAAAGAAAAGAAAAACAA
CAAGAAAGAAAAAAAAAAAA
GAAAAAAAAAAGAAAAAGAA
AAAAAAAGAAAGGAAAAGAA
AAGAAAAGAAAAGAAAAGAA
GAAAGAAAAAAAAAAGACAG
AAGAAAAAAAAGGAAAAGGA
GAAAGAAATAATGAAGAGAA
AAAAAAAAAAAGAGAAAGAA
AAAAAAAAAAAGAAAAAGAA
AAAGGAGGAAAAAACAAAAG
GAAAAAACAAAAGAAAACAT
AAAAGATATAAAAGAGGAAA
CAAAAAAAAAAAGTAGAGGC
GAAAAAGAAAAAGAAGGGGA
CAAAAAATAAATGGCAACAA
GAAAAAAACAAAAGAAAACA
GAAAGATATAAAAGAGGAAA

Figure 5 and 6. On the left are motifs found by GLAM in L. major, and those in T. brucei are on the right. They represent the majority of the motifs found by GLAM in the respective genomes.

MITRA results were more varied than those found by AlignACE or GLAM, but were many were **NTNT** or **NTTNTT** type of repeats.

	A	C	T	G	N	
1	0.25	0.25	0.25	0.25	0.00	*
2	0.50	0.25	0.25	0.00	0.00	A
3	0.00	0.00	0.75	0.25	0.00	T
4	0.25	0.00	0.25	0.25	0.25	*
5	0.25	0.25	0.25	0.00	0.25	*
6	0.25	0.00	0.75	0.00	0.00	T
7	0.25	0.25	0.25	0.25	0.00	*

Figure 7. WMM of a MITRA result

Repeats of this type were not found by AlignACE or GLAM, since AlignACE and GLAM are both Gibb's sampling techniques, while MITRA is a mismatch tree algorithm.

5. Conclusion

From the data we gathered, we found that the **CACA** and **GAGA** repeating motif pattern is perhaps the most likely of the motifs generated to be a functional motif. This is because of its presence in both L. major and T. brucei. It is possible that in addition to the synteny of coding regions, some non-coding regions of importance, such as functional motifs, maybe be conserved among the Trypanosomes. There were many, many other motifs found, and its very likely many of those are functional motifs, but given the scope of this paper we were unable to fully explore how common the less repetitious motifs were among the genome and algorithm results.

Given time, it would be interesting to connect the motifs to the functional classification of the genes within L. major and T. brucei. Many of the genes have BLAST similarity to other proteins with known function, and if that knowledge were integrated it would be possible to categorize upstream regions first by their gene function and then create motif predictions based on that. This would help refine the motif search. Because we utilized a general motif-search strategy naïve of gene function, we probably missed some motifs meant for only a small subset of genes, whose signature was subsumed by more general patterns.

Additionally, as the data becomes available, it would be worthwhile to include other parts of the genome into the research, and perhaps other genomes altogether. L. infantum and T. cruzi also have a high level of synteny between L. major and T. brucei, and the addition of there genes may help to generalize the search for common motifs between the Trypanosome species.

References:

1. Myler, P., Audleman, A., deVos, T., et al. ***Leishmania major*** **Friedlin chromosome 1 has an unusual distribution of protein-coding genes**. Proc. Nat'l. Acad. Sci. USA 1999
2. McDonagh, P., Myler, P., Suart, K. **The unusual gene organization of *Leishmania major* chromosome 1 may reflect novel transcription processes**. Nucleic Acids Research, 2000 Vol. 28, No. 14
3. Eskin, E. and Pevzner, P.A. **Finding Composite regulatory patterns in DNA sequences**. Bioinformatics, 18:1, 354-363. 2002.
4. Frith, M.C., Hansen, Ulla., Spouge, J.L., and Weng, Z. **Finding Functional sequence elements by multiple local alignment**. Nucleic Acids Research, 32:1, 189-200. 2004.
5. Hughes, J.D., Estep, P.W., Tavazoie, S., and Church, G.M. **Computational identification of Cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae***. J.Mol.Biol. 296, 1205-1214. 2000.
6. Lawrence, C.E., Altschul, S.F., et al. **Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment**. Science, 262, 208-214. 1993.