CPSC 545 Project Proposal: Improvement of the PROJECTION Motif Finding Algorithm

Mohammed Alam, Warren Cheung,
Juan Estrada, James King
{malam,wcheung,estrada,king}@cs.ubc.ca

October 6, 2003

1 Introduction

Given a number of DNA sequences, the motif finding problem is the task of discovering a particular polymer that appears (perhaps in a slightly mutated form) in every given sequence. We are considering the problem as defined by Pevzner and Sze[PS00], and as stated by Buhler and Tompa[BT01]:

> **Planted (l, d)-Motif Problem:** Let M be a fixed but unknown nucleotide sequence (the motif consensus) of length l. Suppose that M occurs once in each of t background sequences of common length n, but that each occurrence of M is corrupted by exactly d point substitutions in positions chosen independently at random. Given the t sequences, recover the motif occurrences and the consensus M.

Much work has been done on the problem, but there are certain variations of the problem for which satisfactory solutions have not yet been found. For example, the problem becomes harder if insertions/deletions are considered to be legal mutations or if the motif does not occur in all of the t background sequences.

2 Existing Work

Many motif finding algorithms have been developed. Before 2000, there were essentially two types of algorithms. One kind finds a motif that maximizes a score function representative of how likely the motif is to be 'planted', rather than just randomly occurring. Unfortunately these algorithms often stop at local maxima, completely ignoring absolute maxima that can have much higher scores. The other kind of algorithm is enumerative. Those algorithms are guaranteed to find the most probable motif, but running times become prohibitively slow for larger motifs and more mutations.

Pevzner and Sze [PS00] introduced two new algorithms that were more successful than previous attempts. WINNOWER treats every occurring l-mer as a node in the graph, with nodes adjacent if and only if they differ in at most $2d$ positions and they occur in different sequences. It then finds cliques of size t and works from those starting points. SP-STAR tries testing every occurring l-mer in turn and considers the probability that it is a mutated occurrence of the motif consensus. Using this technique it essentially does an enumerative search, but only over the occurring data rather than the entire space of 4^l l-mers.

In 2002, Buhler and Tompa [BT01] introduced their PROJECTION algorithm. This algorithm 'projects' every occurring l-mer onto a smaller space by hashing. The hash function is based on k of the l positions that are selected at random when the algorithm begins. l-mers are hashed into the same bucket if they have the same bases in those k positions.

The idea behind PROJECTION is that background l-mers (essentially random noise) will be distributed evenly between the buckets. Meanwhile, the 'planted bucket' (i.e. the bucket into which the motif consensus would be hashed) will have additional l-mers because some occurrences of the planted motif will not be mutated in any of the k hashable positions. PROJECTION performs this hashing, then performs refinement on each sufficiently full bucket to find the best motif in that neighbourhood. The algorithm will find the consensus motif if it ends up refining the planted bucket. By running the algorithm multiple times, PROJECTION will refine the planted bucket in at least one run with high probability.

PROJECTION performs significantly better than other algorithms, especially for harder instances of the problem such as $(14, 4)$, $(17, 5)$, and $(18, 6)$. For this reason, we have decided to concentrate on finding improvements for PROJECTION, which has found success in the l-mer hashing technique and, for now, seems to be the right direction to work in for the motif finding problem.

3 Proposed Improvements

The majority of PROJECTION's running time is taken up by the refinement stage that finds the best motif in the neighbourhood of a given bucket. For this reason, it would be advantageous to refine as few buckets as possible in our search for the planted bucket. The difficulty is that refining more buckets is currently advantageous in that it increases the probability that we refine the planted bucket at some point.

Our goal is to find improvements to the projection process in a way that will, in all likelyhood, make projection more complicated and time consuming, but will allow us to refine few enough buckets that the time gained in refinement will more than make up for the time lost in projection.

One problem with the projection method is that some mutated occurrences of the motif consensus will be hashed to buckets other than the planted bucket, essentially being thrown away and considered as noise. In reality, a significant number of motif occurrences will land near, but not in, the planted bucket. Rather than simply refining any bucket with at least some threshold number of l-mers in it, we propose to introduce a more sophisticated bucket scoring system.

We will implement a scoring technique that we call 'bucket aggregation'. A bucket will receive a score based on the number of l-mers projected to it, as well as the number of l-mers that are projected to nearby buckets (diminished by some probability coefficient). Essentially, the neighbour N of a bucket B will count towards B's score in an amount proportional to the probability that an l-mer belonging in B could mutate to land in N. To keep the running time reasonable, we will have to limit the number of neighbours we consider, probably only considering buckets within a Hamming distance of 1 or 2. After our aggregation, we will send to refinement any bucket with a score above some threshold.

By aggregating the scores of nearby buckets, we hope to focus the effect of the significant data (planted motif occurrences) in relation to the noise (randomly occurring l-mers). By doing this, we should improve the chances of sending the planted bucket to refinement. This means that we can send fewer total buckets to refinement and the algorithm will therefore run faster.

The aggregation may also improve Projection's performance in the variation of the problem in which the motif is not planted in all t of the given sequences. That version of the problem is essentially the same but with more added noise. Focussing the significant data may cut through that extra noise to make Projection more effective.

For this project, we will add our aggregation implementation to the existing code, which we have already obtained from `www.cse.wustl.edu/~jbuhler/projection.html`. Along with detailed probabilistic analysis of our method, we will provide experimental analysis. We will compare the success of our augmented version of PROJECTION to that of the original version in a number of different situations. If our modification does not yield any improvement, we plan to give a detailed explain of why we failed.

4 Work Schedule

10.5-10.12: Analysis of existing implementation, including gathering of runtime data and success rates. Probabilistic analysis of bucket aggregation. More in depth research of other algorithms under recent or current development.

10.12-10.26: Implementation, including debugging and testing. Further theoretical analysis. Writing up of bucket aggregation analysis. Ongoing research into current algorithms.

10.26-11.9: Experimentation, experimental analysis. Writing up of comparative experimental results.

11.9-11.16: Editing, proofreading, completion of written paper.

5 Distribution of Work

Mohammed Alam: Coding, testing, (experimentation, related research, writing).

Warren Cheung: Coding, testing, (experimentation, analysis, writing).

Juan Estrada: Related research, experimentation, (writing).

James King: Analysis, writing.

References

[BT01] Jeremy Buhler and Martin Tompa. Finding motifs using random projections. In Thomas Lengauer, David Sankoff, Sorin Istrail, Pavel Pevzner, and Michael Waterman, editors, *Proceedings of the Fith International Conference on Computational Biology (RECOMB-01)*, pages 69–76, New York, April 22–25 2001. ACMPress.

[PS00] Pavel A. Pevzner and Sing-Hoi Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In Russ Altman, L. Bailey, Timothy, Philip Bourne, Michael Gribskov, Thomas Lengauer, and Ilya N. Shindyalov, editors, *Proceedings of the 8th International Conference on Intelligent Systems for Molecular (ISMB-00)*, pages 269–278, Menlo Park, CA, August 16–23 2000. AAAI Press.