

Optimization of sequence distance estimation algorithm parameters with discrete SPSA

George Coulouris

7 May 2009

Abstract

An algorithm to predict Smith-Waterman local alignment scores in linear time using binary frequency vectors is presented. In order to maximize prediction ability, discrete SPSA was used to estimate the optimal algorithm parameters. The optimization process employed Monte Carlo random sequences and the results were validated using real sequences. The score correlation using real sequences was 0.90 with a ROC score of 0.745. The resulting algorithm may be suitable for realtime alignment visualization on commodity graphics processing units.

1 Introduction

Biological sequences with a common structure or function are said to be homologous. While it is frequently of interest to determine whether two nu-

cleic acid or protein macromolecules are homologs, it is currently intractable to predict functional similarity using sequence information alone. Quantifying the extent to which the sequences themselves are similar, however, can serve as a useful proxy for putative biological homology. The mapping of similar regions between sequences is called an alignment, and the quantification of the degree of sequence similarity results in an alignment score.

In some applications, a mapping that covers both sequences in their entirety is desired, even if the sequences are of different lengths; such a mapping is called a global alignment. Conversely, allowing arbitrary subsequences to align to each other yields what is called local alignment. Due to evolutionary differences, local alignments are frequently desired when dealing with experimental biological data. The parameters used to score an alignment are known as a scoring system. Given a scoring system, the score of an optimal local alignment can be determined exactly using the Smith-Waterman (SW) dynamic programming algorithm [8]. Since the quadratic running time of the SW algorithm is prohibitively expensive, heuristics and approximation algorithms are used in practice.

A large class of such approximation algorithms employ so-called n-gram techniques, which first find short regions of exact alignment called seeds; these seeds are then used to grow larger alignments. The Basic Local Alignment Search Tool (BLAST) [1], developed by the National Center for Biotechnology Information (NCBI), is one such algorithm, and is widely considered the de facto standard for computing local alignments. While n-gram tech-

niques operate entirely in the spatial domain, other algorithms employ spectral techniques. Some algorithms utilize the Fourier transform [5]; other algorithms employ so-called k-mer counting techniques [2] [9], which compare histograms of the occurrences of substrings of length k. Such spectral algorithms are typically used to study protein similarity, but there have been recent attempts to apply these methods to nucleotide sequences [4].

2 Motivation

Due to implementation issues, the working set of n-gram algorithms often exceeds the size of the cache on contemporary processors. Since k-mer counting algorithms rely on histograms, they discard some of the spatial information, reducing the amount of data that must be stored. As the value of k grows large, the expected number of occurrences of a particular substring of length k grows small [7]. It follows that, for sufficiently large values of k, binary histograms may be used without losing much information, further reducing the amount of data that must be stored.

Generating a binary k-mer histogram is equivalent to a sequence of bit set operations. Similarly, given two binary k-mer histograms, computing the number of common k-mers is equivalent to counting the number of set bits in the intersection of the histograms. Such an algorithm has linear asymptotic complexity. Since only bitwise operations are used, the binary k-mer histogram algorithm can run efficiently on simple hardware.

I hypothesize that, with the proper choice of parameters, k-mer counting techniques using binary histograms can be used to evaluate DNA sequence similarity. Specifically, I will show that the score approximation determined by k-mer counting correlates with the optimal score determined by dynamic programming.

In order to maximize this correlation, it will be necessary to optimize the parameters of the k-mer counting algorithm. Since it is not practical to evaluate the algorithm at every possible combination of parameters, it is necessary to apply numerical optimization methods. Parameter optimization will be conducted using Monte Carlo random sequences and final validation and evaluation will be conducted using real sequences. Due to noise inherent in the Monte Carlo process, stochastic optimization techniques must be used.

3 Monte Carlo simulation

3.1 Sequence generation

Each evaluation of the loss function involves computing the correlation between k-mer scores and SW scores. Each evaluation requires the generation of n pairs of sequences. Each pair is made of a randomly-generated sequence and a mutated version of the same sequence. Mutations are modeled by point mutations and point insertions/deletions (indels). Non-coding regions of DNA, known as introns, were also simulated by inserting runs of random

sequence. Mutation and indel probabilities of 15% and 1%, respectively, yielded sequences with realistic alignment properties.

3.2 Score correlation

The exact Smith-Waterman score and k-mer score approximation is computed for each sequence pair. After log-transforming both scores, the score correlation is then defined as the correlation coefficient of all sequence pairs examined.

3.3 Noise characteristics

The noise associated with measuring this correlation is approximately normally distributed (Figure 1) and decays as $\frac{1}{\sqrt{n}}$ (Figure 2). This is a direct result of the Central Limit Theorem, which implies that

$$\sqrt{k}(\bar{X}_k - \mu) \xrightarrow{\text{dist}} N(0, \Sigma), k \rightarrow \infty$$

[10]. Choosing $n = 75$ yielded acceptable runtimes at approximately 5% noise.

3.4 Implementation notes

All simulations were performed using the MATLAB Bioinformatics Toolbox. Further tests were performed using an application written with the

Figure 1: Normal plot

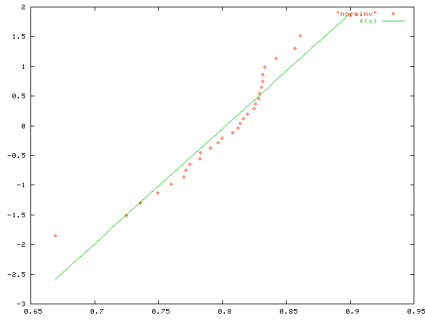
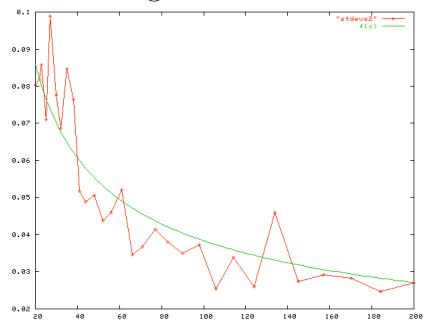


Figure 2: Noise standard deviation vs. # of MC iterations



NCBI C++ Toolkit.

4 Parameter optimization

4.1 Parameter selection

The parameters to be optimized are the word size and the sequence length. Increasing the word size increases the amount of spatial information at the expense of lower resolution [7], while increasing the sequence length increases the density of the histogram. Intuitively, the histograms contain no information if they are completely empty or completely saturated.

4.2 Loss function

In order to cast this optimization problem as a minimization problem, the loss function is defined to be the negative of the score correlation defined above.

4.3 Algorithm selection

A priori, it was not known whether the loss function was either well-behaved or multimodal. A rough exploration of the space revealed a convex, bowl-like shape. Despite its low dimensionality, since the problem exhibits some structure and regularity, SPSA was chosen over a random search algo-

rithm [10].

4.4 Gain selection

Due to the discrete nature of this problem, I implemented SPSA with fixed gains [3]. Since the parameters being optimized are on different scales numerically, I chose to use vector gains. Using the Monte Carlo noise estimates, the SPSA “c” gains were chosen such that Bernoulli perturbations landed on valid grid points; the slope at two grid points was used as a proxy for the gradient. The SPSA “a” gains were then chosen manually to give good performance.

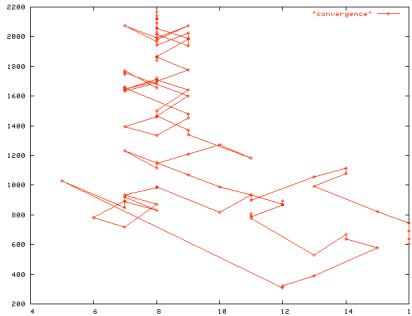
4.5 Constraints

From the standpoint of software implementation, practical values of the word size range from approximately 4 to 16. While no such restrictions apply to the tile size, they were bounded above by 4000 to keep the Monte Carlo running time manageable. In the context of SPSA, these limitations were implemented as hard hypercube constraints.

4.6 Global vs. local

Restarting the optimization process multiple times from different areas of the search space converged to similar results (Figure 3). While these empirical observations do not prove that better solutions do not exist elsewhere,

Figure 3: Example SPSA realization



particularly given the constraints, they provide some confidence that there is a single global optimum in the domain of interest.

4.7 Convergence

The SPSA process was allowed to run for 50 iterations per replication. Typical realizations approached the optimum after 25 iterations.

4.8 Results

Running 30 independent replications of the discrete SPSA process gave a mean initial loss value of 0.59 and a mean terminal loss value of 0.78. The standard deviations were approximately equal at $s=0.065$. A t-test easily showed ($t = 11.38$, $p = 2.2 \times 10^{-12}$) that the optimization process significantly improved the value of the loss function. Due to the discrete nature of the problem, it is not appropriate to compute a mean parameter vector; computing the medians instead yielded a word size of 9 and a sequence length

of 2254.

5 Validation

5.1 Data used

Using the parameters obtained from SPSA optimization, the binary k-mer frequency algorithm was tested on a set of 143 sequence pairs. The first sequence of each pair is a human mRNA sequence drawn from the NCBI Reference Sequence Database (RefSeq)[6]. The second sequence of each pair is a non-human mRNA that was shown to have a significant alignment to the first sequence with BLAST. The sequences were chosen such that there are no significant alignments between sequences that are not in the same pair.

5.2 Score correlation

The score correlation improved from 0.78 for randomly-generated sequences to 0.90 for real sequence pairs (Figure 4), despite the fact that the real sequences were not of uniform or optimal length.

5.3 ROC curve

The k-mer scoring algorithm can be used to answer classification problems by applying a score threshold. The number of true positives can be evaluated

Figure 4: Score correlation

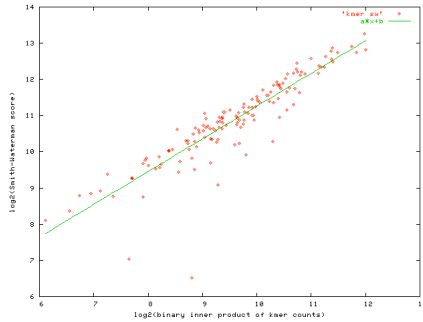
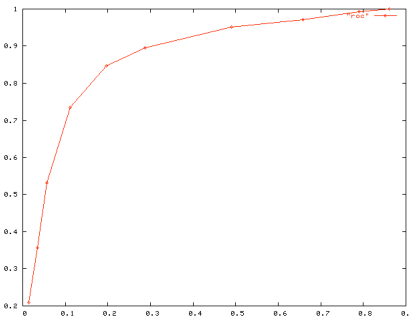


Figure 5: ROC curve



by running the k-mer algorithm on the sequence pairs that are known to contain alignments; similarly, the number of false positives can be found by running the k-mer algorithm on the sequence pairs that are known not to contain alignments. Repeating this process for various values of the threshold results in a plot of true positives vs. false positives, commonly known as a Receiver Operating Characteristic (ROC) curve. The ROC score, defined as the area under the curve, is 0.745 (Figure 5). A score threshold of 8 resulted

in a true positive rate of 0.89 and a false positive rate of 0.28.

6 Conclusion

Discrete SPSA converged to optimal parameters for the binary k-mer counting algorithm. Using these parameters, the binary k-mer counting algorithm was able to predict the score of optimal local alignments in real sequences. While insufficiently accurate for rigorous sequence analysis work, the implementation simplicity of such methods makes them promising for realtime alignment visualization.

7 Future work

Since fixed gains are used, SPSA can never truly converge to the true optimum. It would be reasonable to divide the optimization into two phases, the first phase consisting of 25 iterations that drive to the optimum, and another 25 iterations used for the purposes of iterate averaging. It may also be desirable to optimize the ROC score directly by using it as the loss function during parameter optimization.

References

- [1] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new

- generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, September 1997.
- [2] R. C. Edgar. Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucleic Acids Res*, 32(1):380–385, 2004.
- [3] L Gerencser, Stacy D. Hill, and Zsuzsanna V. Optimization over discrete sets via SPSA. In *Winter Simulation Conference*, pages 466–470, 1999.
- [4] T. Kahveci, V. Ljosa, and A. K. Singh. Speeding up whole-genome alignment by indexing frequency vectors. *Bioinformatics*, 20(13):2122–2134, September 2004.
- [5] Kazutaka Katoh, Kei-ichi Kuma, Hiroyuki Toh, and Takashi Miyata. MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucl. Acids Res.*, 33(2):511–518, 2005.
- [6] K. D. Pruitt, T. Tatusova, and D. R. Maglott. Ncbi reference sequences (refseq): A curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*, 35(Database issue), January 2007.
- [7] Gesine Reinert, Sophie Schbath, and Michael S. Waterman. Probabilistic and statistical properties of words: An overview. *Journal of Computational Biology*, 7:1–46, 2000.

- [8] T. F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197, 1981.
- [9] Erik Sonnhammer and Volker Hollich. Scoredist: A simple and robust protein sequence distance estimator. *BMC Bioinformatics*, 6(1):108, 2005.
- [10] James C. Spall. *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 2003.