APPLICATION OF OPTIMIZATION ALGORITHMS TO THE GENOME ASSEMBLY

Robin Jugas

Doctoral Degree Programme (2.), FEEC BUT

E-mail: jugas@feec.vutbr.cz

Supervised by: Helena Škutková

E-mail: skutkova@feec.vutbr.cz

Abstract: The paper results from development of new sequencing methods together with the need of suitable genome assembly algorithms. It combines the genomic signal processing, correlation techniques and optimization algorithms for solving assembly task. Genomic signals are made by conversion of letter-based DNA into the form of digital signal, thus the methods of digital signal processing can be applied. Possible overlaps between reads converted into signals are found by computing correlation coefficient similarly to cross-correlation. We acquire similarity matrix and the task is to find the path through it achieving minimum distance criterion. For the task, the two optimization techniques were employed: ant colony optimization (ACO) and simulated annealing (SA). The result implies the possibility of using the ACO at the task of creating path through similarly to graph-theory-based algorithms.

Keywords: bioinformatics, genome assembly, genomic signal processing, optimization techniques

1 INTRODUCTION

Genome assembly and DNA sequencing methods are stressed because of medical trends focusing on a genetic base of diseases and possible treatment [1]. Thus, new methods of molecular biology are developed and also a price of fully sequenced genome decreased, making progress more available [2]. However, DNA sequencing methods still are not able to sequence the whole genome at once, hence the need for a genome assembler. An output of DNA sequencers differs in amount and length of reads – a short sequences of DNA which are later connected into longer sequences using genome assembler. Older platforms like Roche454 produced up to 1000bp reads, Illumina up to 350bp, recent Oxford Nanopore can sequence thousands of base pairs [3]. More assemblers exist targeting different properties of sequencing platforms [4]. For long reads, the algorithms based on overlap-layout-consensus graphs prove to be efficient [5], whereas for shorter reads the assemblers are using de Bruijn graphs [6]. Assemblers use the graph theory algorithms for searching a path through reads, which shares an overlap between pairs of them.

All assemblers use the standard letter-based form of DNA, using four letters denoting nucleotides. The current study focusses on Genomic Signal processing(GSP) techniques to improve the previous existing methods. [7]. Genomic signal processing merges the domains of bioinformatics and digital signal processing. GSP is based on numerical representations, formulas describing the conversion of letter-based DNA into numerical based DNA. After the conversion, methods of digital signal processing can be used.

The purpose of this article is to examine the possibility of using optimization techniques – ant colony optimization (ACO) and simulated annealing (SA), at the genome assembly. Genome assembly can be viewed as searching through a graph and creating a path. Both selected optimization techniques are suitable for graph problems and tend to find global extreme. [8]

2 METHODS

2.1 GENOMIC SIGNAL PROCESSING

Genomic signal processing combines the genomic signals with the field of digital signal processing. The genomic signal is generally a signal obtained from DNA sequence by applying certain formulas. The output differs in the dimensionality of a signal, added features or possibility of backward reconstruction of the sequence. The method for transforming DNA into the signal is called numerical representation and several of them were already published [9, 10]. Suitable numerical representations are those one-dimensional with the same sampling frequency as the original sequence, thus, the phase numerical representation is chosen [11]. Each nucleotide in DNA sequence is converted into a genomic signal by getting the phase of the complex number according to mapping, e.g. (A = π/4; T = -π/4; C =-3π/4; G =3π/4) and then the values are cumulatively summed [11].

2.2 CORRELATION COMPUTATION

Searching for overlaps between reads is based on correlation methods – Pearson correlation coefficient is computed between all possible pairs of reads using implemented gradually incrementing window. The Pearson correlation is defined as (1):

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\frac{(A_i-\mu_A)}{\sigma_A}\frac{(B_i-\mu_B)}{\sigma_B} \quad (1)$$

where *A*, *B* denote signals, N is the length of signals, μA and σA are the mean and standard deviation of *A*, respectively of *B*. Pearson correlation coefficient values are in range from -1 to 1, a value of 1 indicates linear dependency of vectors A and B, a zero value indicates no dependency and a value of -1 negative linear dependency. For searching overlaps, i.e. detecting the position where the signals share a part of them, the Pearson correlation is computed using an algorithm similar to the cross-correlation method. Correlation computation is divided into two steps, so all possible positions between reads of the same length are covered. Having two signals, *X [1: N]* and *Y [1: M]*, *N=M*, the correlation coefficient is calculated as (2):

$$coef_{a)} = \rho(X\,[1:i], Y\,[M-i:M]) \qquad coef_{b)} = \rho(X\,[N-i:N], Y\,[1:M-i]) \quad (2)$$

where ρ is the Pearson correlation coefficient and i is the shift between the signals. The $coef_{a)}$ describes the computation when the first signal is gradually shifted into the second one to the right side, the $coef_{b)}$ describes the same from the other side, the first signal is shifting out of the second signal. The formulas are graphically described in Figure 1.

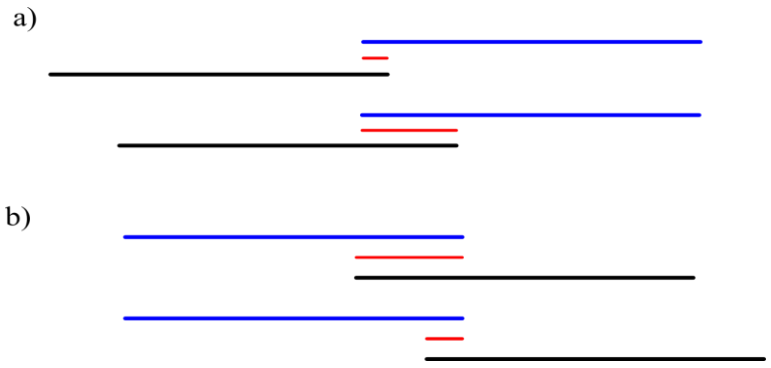

Figure 1: Algorithm for computing correlation. Blue and black lines represent signals, red lines show what part of a signal are taken into correlation computation

The output of this calculation is correlation signal of length *2N*, where axis x denotes shifted position of signals and y denotes correlation coefficient at that point. There should be significant extreme at a position where signals have overlap. Hereby we acquire a similarity matrix by choosing the maximum value of correlation signal between a pair of reads.

2.3 SIMULATED ANNEALING

Simulated annealing is a probabilistic method for searching global optimum which is based on metallurgy process of creating crystal lattice with minimal energy state. Similarly, the algorithm searches for lower energy state. The current state represents one possible solution, in traveling salesman, it is the permutation of cities to visit and algorithm considers neighbouring state, which is generated by reversing the order of cities. The simulated annealing explores the possible solutions by using the neighbours state and it can accept currently worse neighbour state with a certain probability (3):

$$r \leq e^{\left[fp_{old} - fp_{new}\right]/T} \tag{3}$$

where r is uniform random number and T denotes temperature. Otherwise, the new solution is rejected.

Key role plays the temperature T, higher T implies greater random fluctuations of neighbouring states. Decreasing of T is called cooling schedule. The algorithm stops when T≈0.

2.4 ANT COLONY OPTIMIZATION

Ants are finding the shortest path to food by following the pheromone trail left by other ants. Ants will exude more pheromone on a short path than on long path. All possible ways are visited with the same probability, but the ants going through the shortest will exude the pheromone faster than others. Because of premature convergence, the property of pheromone evaporation was added, so the initially strong pheromone paths can be later abandoned. Each ant is visiting cities according to probability (4):

$$p_{mn}^{k} = \frac{\tau_{mn}^{a}/d_{mn}^{b}}{\sum_{q} \tau_{mq}^{a}/d_{mq}^{b}} \tag{4}$$

where τ is pheromone strength, *q* cities on tour *k* after city *m*, *a* is pheromone weighting, *b* is distance weighting. Pheromone is laid between cities according to formula (5):

$$\tau_{mn} = (1 - \xi)\tau_{mn} + \sum_{k=1}^{Nants} \tau_{mn}^{k} + \varepsilon\tau_{mn}^{elite} \tag{5}$$

where τ_{mn}^{k} is pheromone between cities m and n laid by ant k, $\xi$is pheromone evaporation constant, ε is elite path weighting constant, τ_{mn}^{elite} is pheromone laid to this point.

Short path with higher pheromone levels has higher probability to be selected. The algorithm ends after fixed iteration or after achieving selected criteria.

2.5 DATASET AND EVALUATION METRICS

A synthetic dataset was created for purpose of evaluating accuracy of employed methods. All reads have a length of 350bp likewise Illumina output. Between the reads are overlaps of defined length: 25, 50, 75, 100 nucleotides. Finally, three datasets of 50, 100 and 200 reads we created. Reads were created from the whole genome sequence of E. coli (GenBank ID NC_000913.3). The output is evaluated as correctly or incorrectly joined pair of reads. Thus, used metrics are the percentage of correctly and incorrectly joined reads.

2.6 RESULTS

50 reads			100 reads		200 reads	
Overlap length	% JNS	% DJNS	% JNS	% DJNS	% JNS	% DJNS
25	100	0	100	0	100	0
50	100	0	100	0	100	0
75	100	0	100	0	100	0
100	100	0	100	0	100	0

Table 1: Results of ant colony optimization; %JNS – correct joins, %DJNS-incorrect joins

50 reads			100 reads		200 reads	
Overlap length	% JNS	% DJNS	% JNS	% DJNS	% JNS	% DJNS
25	73,8	26,1	56,3	43,6	49,5	50,4
50	73,4	26,5	59,5	40,4	50,3	49,6
75	73,4	26,5	62,6	37,3	50,1	49,8
100	75,5	24,4	61,4	38,5	51,1	48,9

Table 2: Results of simulated annealing; %JNS – correct joins, %DJNS-incorrect joins

3 DISCUSSION

The aim of the paper was to examine the applicability of optimization techniques for genome assembly paired with genomic signal processing. Two optimization techniques were employed: ant colony optimization and simulated annealing. The results were evaluated at the dataset considering different length of overlap together with increasing number of reads considered. The results are placed into Tables 1 and 2. The algorithm for correlation computation was able to detect precisely all overlaps, however only the ant colony optimization was proven to successfully create a path through similarity matrix. ACO achieved maximum score in all cases. However, algorithm of simulated annealing did not perform accurately. With increasing number of reads, its accuracy decreased up to 50 %. Also, there is a tendency for a higher accuracy towards a longer reads, which did not exhibit by ACO. On the other hand, the computation of ACO took considerably longer time. State-of-the-art assemblers are using graph theory-based algorithm for creating a path through reads, another way of using suitable optimization technique was proven possible.

ACKNOWLEDGEMENT

This work was supported by grant project GACR 17-01821S. Computational resources were partially provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures".

REFERENCES

[1] G. M. Frampton, A. Fichtenholtz, G. A. Otto, et all, “Development and validation of a clinical cancer genomic profiling test based on massively parallel DNA sequencing,” Nat. Biotechnol., vol. 31, no. 11, pp. 1023–1031, Nov. 2013.

[2] D. Deamer, M. Akeson, and D. Branton, “Three decades of nanopore sequencing,” Nat. Biotechnol., vol. 34, no. 5, pp. 518–524, 2016.

[3] S. Koren and A. M. Phillippy, “One chromosome, one contig: Complete microbial genomes from long-read sequencing and assembly,” Curr. Opin. Microbiol., vol. 23, pp. 110–120, 2015.

[4] J. T. Simpson and M. Pop, “The Theory and Practice of Genome Sequence Assembly,” Annu. Rev. Genomics Hum. Genet., vol. 16, no. 1, pp. 153–172, 2015.

[5] E. W. Myers, “A Whole-Genome Assembly of Drosophila,” Science (80-.)., vol. 287, no. 5461, pp. 2196–2204, 2000.

[6] D. R. Zerbino and E. Birney, “Velvet: Algorithms for de novo short read assembly using de Bruijn graphs,” Genome Res., vol. 18, no. 5, pp. 821–829, 2008.

[7] P. Ramachandran and A. Antoniou, “Genomic Digital Signal Processing,” Parameswaran;andreas antoniou.

[8] R. L. Haupt and S. E. Haupt, Practical Genetic Algorithms, 2nd ed. Wiley, 2005.

[9] P. D. Cristea, “Conversion of nucleotides sequences into genomic signals.,” J. Cell. Mol. Med., vol. 6, no. 2, pp. 279–303, Apr. 2002.

[10] M. Abo-Zahhad, S. M. Ahmed, and S. a. Abd-Elrahman, “Genomic Analysis and Classification of Exon and Intron Sequences Using DNA Numerical Mapping Techniques,” Int. J. Inf. Technol. Comput. Sci., vol. 4, no. 8, pp. 22–36, 2012.

[11] P. D. Cristea, “Phase analysis of DNA genomic signals,” Proc. 2003 Int. Symp. Circuits Syst. 2003. ISCAS ’03., vol. 5, p. V-25-V-28, 2003.