SBARS: fast creation of dotplots for DNA sequences on different scales using GA, GC-content

Pyatkov, M.I. †, Pankratov, A.N.
Institute of Mathematical Problems of Biology RAS, Pushchino, Russia.

Associate Editor: Prof. Alfonso Valencia

1 INTRODUCTION

The vast majority of approaches used to analyze nucleotide and amino acid sequences are based on algorithms working with text strings. Until recently, such methods were justified since the length of processed genetic text was relatively short. The evolution of sequencing techniques, however, has resulted in dramatically increased data sets, providing nucleotide sequences that are comparable to whole genome in size. Algorithmic “correction” of point mutations is computationally intensive and time consuming. At the same time it is reasonable to assume that a number of mutations are unintentionally incorporated into comparisons of large fragments of DNA (more than 10,000 bp) resulting in significant decrease in efficiency of the text-based algorithms.

To bypass the problems of the text-based algorithms, a number of effective spectral algorithms based on Fourier transform were developed, which are used to search for minisatelites (SRF (Sharma et al., 2004) and OWMSA (Du et al., 2007)), multiple alignments (MAFFT, Katoh et al., 2002) etc. Despite a high performance, these tools are focused on finding short repeats and have a number of limitations for the analysis of long sequences and searching for extended homologous fragments.

In the present article, we propose an original method for finding different types of long repeats in genome-scale DNA sequences. We also offer a possible solution for the problem of dotplot creation whose time complexity can be reduced to $\Theta(n)$.

†to whom correspondence should be addressed

2 METHODS

SBARS is based on the analysis of a function obtained from the nucleotide sequence. In our method we used GC-content curve as a higher order representation of the nucleotide sequence. The main parameters of GC-content are sliding window ($W_1$) and its step ($d_1$). The resulting GC-content curve is divided into overlapping frames with width ($W_2$), wherein each frame is displaced relative to the other by a step $d_2$. After that we perform pairwise comparison of frames by integral estimation of distance between them:

$$
\rho = \frac{1}{W_1^2 W_2} \sum_{i=1}^{W_2} (f_i - g_i)^2,
$$

(1)

where $f_i, g_i$ are two fragments of GC-content. Note that $0 \leq \rho \leq 1$, since $f_i \leq W_1$ and the number of terms in thsum is equal to $W_2$. Therefore, the distance does not depend on the sizes of the windows. For recognition of repeats, the following decision rule is used: if $\rho \leq \varepsilon$ where $\varepsilon$ is a threshold, then the fragments are considered to be similar if $\rho \geq \varepsilon$, the fragments are not similar (Fig. 1a).

In addition to the GC-content, we used the GA-content curve with the same parameters $W_1$ and $d_1$ which allow us to unambiguously recover a DNA sequence from this curves (see Supplementary). Simultaneous recognition by two curves provides more stable results and allows us to define the various types of repetitions with minimal computational cost. For example, consider a fragment of GA-content with length $W_2$, which in the function values by definition is limited by window $W_1$ size. Fig. 1b shows three types of transformation under GA-content, and each of them is related to the corresponding transformation under DNA sequence if this sequence is decoded from GC-GA-content curves. For reversed DNA sequence we made reverse GA-content curve. To obtain the fragment of the DNA sequence corresponding to complementary sequence, the GA-content should be transformed to CT-content using the following expression: CT-content $= W_1 -$ GA-content. For reverse-complement transform one needs to make both transforms, which are described above, in an arbitrary order.

The main feature of the method is that all of the fragments of GC-GA-content are approximated using orthogonal polynomials (Legendre, Chebyshev, Fourier) and are presented in the form of the expansion coefficients. Thus, all the transformations and estimations of expression (1) are performed using the vectors of expansion coefficients, and this allows us to identify similarity between the GC-GA-content fragments by comparing the first few coefficients (usually about 10) (Pankratov et al., 2012).

The other feature of the method is scalability. The dotplot size depends not only on the size of nucleotide sequence but also on the parameters. In common case the complexity of our algorithm can be represented as three components. The first component is making GC-GA-content whose time complexity is linear. The second component is evaluating vectors of expansion coefficients whose complexity is also linear, because of the number of frames to be converted into vectors is $\frac{W_2}{W_1}$. The last component is pairwise comparison of vectors and constructing dotplot. The complexity...
of the last component is $\Theta\left(\frac{n^2}{d^2}\right)$. If the size of a sequence is increasing but parameters are fixed, the complexity of this component is quadratic, and consequently whole complexity will also be quadratic. But if we increase the parameters proportionally to the length of the sequence, then the last component will be constant and complexity of whole algorithm will be $\Theta(n)$. In the last case the dotplot size does not depend on the sequence length. This facilitates construction of low resolution matrix even for large nucleotide sequences.

3 IMPLEMENTATION

SBARS is a complete implementation of the algorithm given above with graphical user interface (GUI). The main objective of the program is to construct a dotplot and obtain the coordinates of repeated fragments in the sequences. The program is an advanced real-time viewer for DNA sequences at different scales. The program is written in C++ with OpenMP directives, and GUI is based on the QT library. A detailed guide with examples is located at: https://github.com/mpyatkov/sbars/raw/master/SBARS.pdf

Table 1. Benchmark results for SBARS and Gepard

<table>
<thead>
<tr>
<th>Sequence length</th>
<th>Gepard</th>
<th>SBARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>70000</td>
<td>&lt; 1 sec</td>
<td>&lt; 1 sec</td>
</tr>
<tr>
<td>5000000</td>
<td>45 sec</td>
<td>1.6 sec</td>
</tr>
<tr>
<td>Human chr. Y</td>
<td>4 min 45 sec</td>
<td>27 sec</td>
</tr>
<tr>
<td>Mus chr. 6 vs Rat chr. 4</td>
<td>45 min</td>
<td>45 sec</td>
</tr>
</tbody>
</table>

Selfplots of sequences with different lengths have been calculated on a 2.2 GHz AMD Phenom 9550 Quad-Core machine using Gepard and SBARS. The resulting dotplots and the corresponding parameters are presented in the Supplementary.

We compared our program with Gepard (Krumsiek et al., 2007), which is the closest equivalent to our program by performance. The time complexity of Gepard is $\Theta(n \log n)$. The output of the programs is fixed size dotplot for sequences in different scales. For similar quality dotplots (see Supplementary), especially for long sequences, our method demonstrates less computation time (Table 1).

4 DISCUSSION

SBARS is a fast and efficient tool for identifying dispersed (direct, inverted) and tandem DNA repeats. The program is not aimed at the comparison of individual nucleotides. The main idea of this approach is to quickly identify the similarity of individual fragments within the query sequences, disregarding single nucleotide insertions or deletions. The current version of the program efficiently identifies repeated sequences and can be developed for the analysis of long insertions or deletions due to chromosome rearrangements in similar sequences. The underlying spectral algorithm demonstrated good scalability on multi-core processors (Pankratov et al., 2010).

ACKNOWLEDGEMENTS

Funding: The work was supported by the Russian Foundation for Basic Research, Grants 14-07-31306, 14-07-00924 and by Intel company, T-Platforms company.

REFERENCES