BIOINFORMATICS APPLICATIONS NOTE Vol. 23 no. 8 2007, pages 1026–1028
doi:10.1093/bioinformatics/btm039

Sequence analysis

Gepard: a rapid and sensitive tool for creating dotplots on genome scale

Jan Krumsiek[1,*], Roland Arnold[1] and Thomas Rattei[2]

[1]Institute for Bioinformatics (MIPS), GSF National Research Center for Environment and Health, Ingolstädter Landstraße 1, 85764 Neuherberg and [2]Technische Universität München, Department of Genome Oriented Bioinformatics, Am Forum 1, 85354 Freising, Germany

Received on October 13, 2006; revised on January 30, 2007; accepted on January 31, 2007

Advance Access publication February 19, 2007

Associate Editor: Martin Bishop

ABSTRACT

Summary: Gepard provides a user-friendly, interactive application for the quick creation of dotplots. It utilizes suffix arrays to reduce the time complexity of dotplot calculation to $\Theta(m^*\log n)$. A client–server mode, which is a novel feature for dotplot creation software, allows the user to calculate dotplots and color them by functional annotation without any prior downloading of sequence or annotation data.

Availability: Both source codes and executable binaries are available at http://mips.gsf.de/services/analysis/gepard

Contact: krumsiek@in.tum.de

1 INTRODUCTION

Dot matrix analysis is a popular method for bioscientists to quickly create complete comparisons of two proteins or nucleic acid sequences. Dotplots, which are the graphical results of dot matrix analysis, can be used to interpret and analyze the evolutionary relationships of the sequences by examining conserved domains, reverse matches and repeats. The principle of dotplots has already been introduced several decades ago (Fitch, 1969; Gibbs and McIntyre, 1970). The 'DOTTER' program was the first interactive dotplot tool with a graphical user interface (Sonnhammer and Durbin, 1995).

The major problem of dotplot calculation is the time complexity of $\Theta(m \cdot n)$, where m is the length of the shorter, n the length of the longer sequence. Already for sequences of subgenome size, this results in unacceptable computation times even on modern computer systems. A possible solution to this problem is the usage of a word index for the quick identification of matching substrings. Several programs already employ such sequence-preprocessing techniques with fixed word lengths (Huang and Zhang, 2004; Szafranski *et al.*, 2006). A suffix-structure-based word lookup approach capable of producing dotplots has already been used in the *MUMmer* software (Kurtz *et al.*, 2004). Our program utilizes the suffix array data structure for quick substring lookups, which reduces the time complexity to $\Theta(m \log n)$. It allows interactive adjustment of the word length to avoid loss of information.

A novel functionality of the Gepard project is a client–server dotplot mode. The server provides all publicly available genomes including functional annotations of the genes. This feature allows the user to rapidly compare different genomes interactively without need to download any data manually.

2 PROGRAM FEATURES

2.1 Interactive, user-friendly interface

Gepard is accessed via a graphical user interface. Selecting the input sequences, configuring the dotplot parameters (e.g. word length or window size) and navigating through the plot are done using the computer mouse. The usage is intuitive and easy to learn, and context-sensitive help is available for all interface items. The program also includes a feature that was first introduced as the 'greyamp tool' in DOTTER. The user can use scrollbars for immediate noise filtering within the dotplot. This helps to determine and emphasize significant parts of the plot with highly similar areas.

2.2 Fast dotplot computation

For large-scale dotplots, the program uses suffix arrays for heuristical dotplot computation. It searches for words of a certain length (10 by default) from the shorter sequence in the suffix array of the longer sequence. This reduces the dotplot computation complexity as described above, but requires matching subsequences of at least the selected word length to generate a hit.

For smaller dotplots or zoomed-in regions, the program switches to the classical window-based dotplot calculation method as in DOTTER. In contrast to DOTTER, our method is also able to detect reverse complementary matches.

2.3 Suffix array calculation

Gepard uses the Skew algorithm for suffix array calculation (Kärkkäinen and Sanders, 2003). The suffix arrays are computed in-memory with optional persistent storage on the hard drive to avoid recalculation.

*To whom correspondence should be addressed.

Due to the recursive method, this algorithm is very space consuming (~25 times the space of the sequence). Therefore, Gepard is able to utilize the program *mkvtree*, which is part of the *Vmatch* package (Abouelhoda *et al*., 2003) (http://www.vmatch.de/). This program uses a faster and more space-efficient algorithm to compute persistent suffix arrays. If this software is available on the executing machine, it is used by Gepard automatically.

2.4 Client–Server mode based on webservice technology

The program includes a 'remote mode' in which the user can select genomic sequence contigs from the PEDANT database (Frishman *et al*., 2003). As no downloading of sequence data is necessary, this provides a convenient way for the quick comparison of all publicly available genomic sequences.

The Gepard client sends a dotplot request to the server. After processing the request, the complete dotplot is immediately transferred back to the client via the network connection. The result is a fully featured dotplot with all filtering options as mentioned above. Another feature of this client–server system is the link-out to PEDANT. When the user clicks in the dotplot, the corresponding gene information page from PEDANT is shown in a web browser.

Gepard's client–server communcation is based on webservice technology. Webservices allow platform-independent communication between computer systems over a network. They usually utilize the HTTP protocol, which yields an important feature: Webservices should be accessible from any public or scholar computer system because communication through the HTTP ports is granted in most firewall systems. The remote functionality is provided by an Axis Webservice installed on an Apache Tomcat application server at http://mips.gsf.de.

2.5 Functional annotations

Gepard integrates the main functional categories of MIPS's Functional Catalogue 'Funcat' (Ruepp *et al*., 2004). This allows the user to color genes in the plot by their functional category. The catalogue includes categories like 'Metabolism', 'Storage protein', 'Transcription', 'Protein synthesis' and 'Cell fate'. The PEDANT database includes functional annotation information for every genome. If available Gepard uses manual annotation data from the database, otherwise the automatically annotated functional categories are displayed.

2.6 Platform-independent binaries

Gepard is implemented in Java. Due to the nature of Java software technology, only one executable is required to run Gepard on any Java-enabled platform (which includes Microsoft Windows™, Mac OS™ and any common Un*x platform). Furthermore, Gepard can be run via Java Webstart, which enables the user to start Gepard without any prior manual downloading or installation directly from within the web browser.

3 RESULTS

We compared dotplot calculation speeds between Gepard and DOTTER. We have chosen this comparison because Gepard is able to achieve the same quality of dotplots as DOTTER. All tests were performed on an Intel Pentium Centrino 1600 MHz machine. The results clearly demonstrate the reduction of time complexity for dotplot calculation in Gepard. Even the computation of genome-scale dotplots in acceptable time becomes possible (Table 1). The dotplot computation time may decrease noticeably if pre-calculated suffix arrays are used. We also compared our program with the suffix-tree-based MUMmer software. Although the calculation time of plots based on unique matches (measured by running MUMmer with –mum) is four times shorter than Gepard, the creation of plots based on all matches (option maxmatch) as required for interactive dotplots took up to 20 times longer compared to Gepard.

Table 1. Benchmark results for DOTTER and Gepard

Sequence length	DOTTER	Gepard	Gepard pre-SA
10 000 bp	2 s	<1 s	<1 s
50 000 bp	30 s	<1 s	<1 s
100 000 bp	2 min 4 s	<1 s	<1 s
1 000 000 bp	2 h 10 min	5 s	4 s
5 000 000 bp	52 h 38 min[a]	47 s	40 s
Human chrom. I	382 years[a]	61 min	53 min

Selfplots of sequences of different lengths have been calculated on a (2.33 GHz Intel Xeon machine) using DOTTER, Gepard without pre-calculated suffix arrays and Gepard utilizing pre-calculated suffix arrays ('Gepard pre-SA').
[a]Extrapolated time value.

Optimal sensitivity can be achieved by adjustable word lengths and window sizes. The information content of the dotplots is enriched by the coloring of regions in the plot by their functional annotation.

Conclusively, Gepard provides a complete and user-friendly software package for the comparative investigation of proteins and nucleic acid sequences. Parameter changing, zooming and filtering are integrated in one single application (Fig. 1). It utilizes a rapid dotplot calculation algorithm and includes many productive features to efficiently work with dotplots. Furthermore, Gepard is the first dotplot tool with server-side, online computation abilities. This allows bioscientists to conveniently compare many pairs of genomic sequences in short time.

4 REQUIREMENTS

The program requires Java 5.0 (or higher) Runtime Environment, which is freely available at http://www.java.com. For webservice usage, a DSL or faster internet connection is strongly recommended.

ACKNOWLEDGEMENT

We thank Volker Stümpflen and Dominik Lindner for their support in setting up the webservice and the anonymous reviewers for the important comments.

Conflict of Interest: none declared.

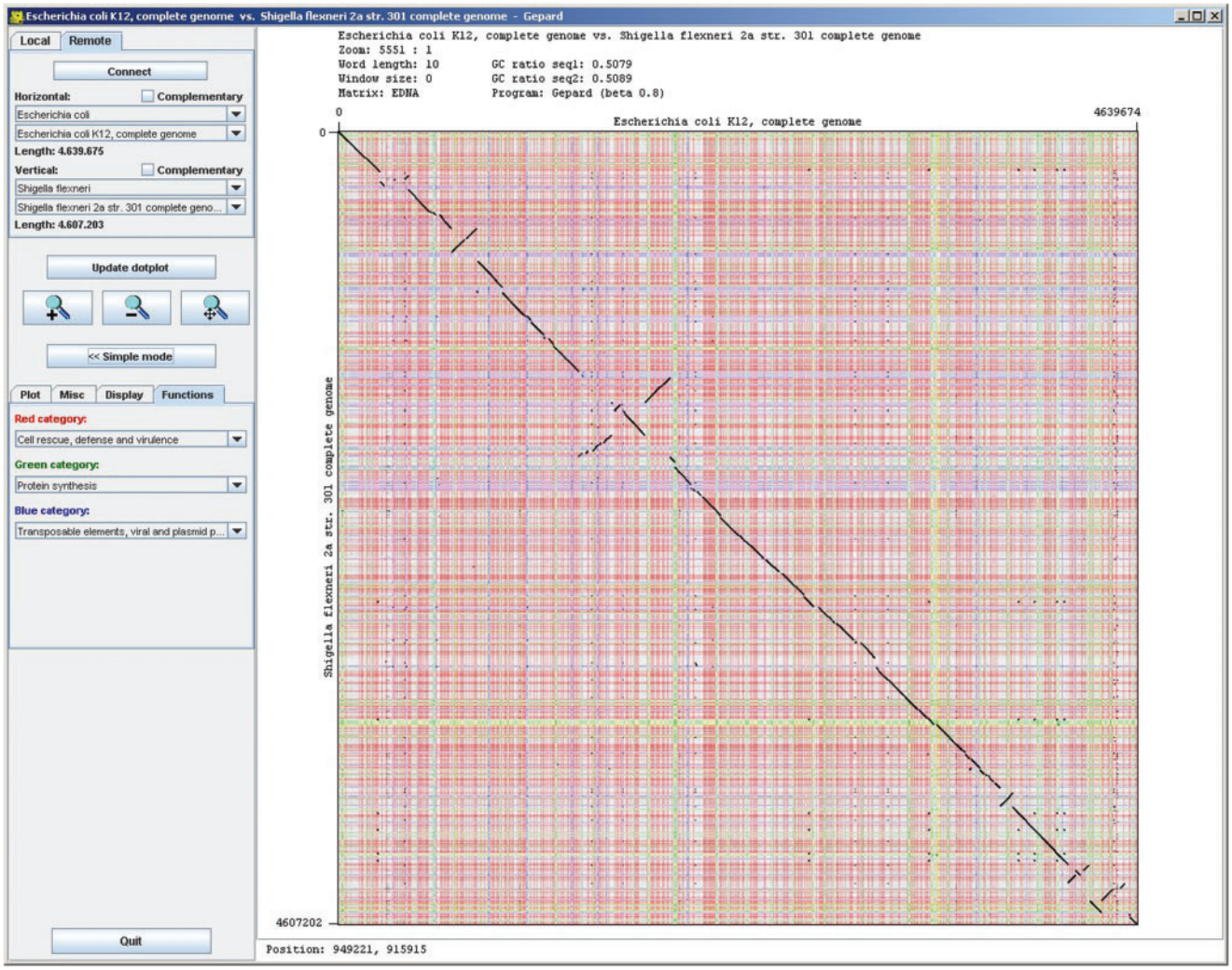

Fig. 1. Gepard application in remote mode displaying a dotplot of *Escherichia coli* versus *Shigelia flexneri* with colored functional annotations.

REFERENCES

Abouelhoda *et al.* (2004) Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, **2**, 53–86.

Fitch,W.M. (1969) Locating gaps in amino acid sequences to optimize the homology between two proteins. *Biochem. Genet.*, **3**, 99–108.

Frishman *et al.* (2003) The PEDANT genome database. *Nucleic Acids Res.*, **31**, 207–211.

Gibbs,A.J. and McIntyre,G.A. (1970) The diagram: a method for comparing sequences. Its use with amino acid and nucleotide sequences. *Eur. J. Biochem.*, **16**, 1–11.

Huang,Y. and Zhang,L. (2004) Rapid and sensitive dot-matrix methods for genome analysis. *Bioinformatics*, **20**, 460–466.

Kärkkäinen,J. and Sanders,P. (2003) Simple linear work suffix array construction. In *Proc. 13th International Conference on Automata, Languages and Programming*. Springer.

Kurtz *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.

Ruepp,A. *et al.* (2004) The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res.*, **32**, 5539–5545.

Sonnhammer,E.L. and Durbin,R. (1995) A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis. *Gene*, **167**, GC1–GC10.

Szafranski,K. *et al.* (2006) tuple_plot: fast pairwise nucleotide sequence comparison with noise suppression. *Bioinformatics*, **22**, 1917–1918.