BLAST organism cross comparison as a tool for genome closure

Lucas B. Chaney

Departments of Computer Science and Biology, Hiram College, Hiram, Ohio 44234 USA

Submitted April 14, 2004 in partial completion of CPSC 401, Professor A. Guercio, Hiram College

ABSTRACT

Motivation: Tremendous effort and interest has recently been applied to the process of genomics, the determination of the entire DNA sequence of an organism of interest to researchers. The development of new methods, as well as the increased demand for this technology, has lowered the cost of sequencing and allowed many smaller academic institutions to become actively involved in genome efforts. As a wide range of new players become involved in these projects, a new tool is also becoming available—the ever enlarging database of known sequences. It has been the goal of the program developed by this project to provide an easy to use interface for facilitating genome completion by combining a number of steps in one simple application and allowing users of any technical level the ability to use BLAST comparison as a tool for gap closure.

Results: We have successfully applied this method to the *Streptococcus sanguis* unfinished genome for suggestion of contig alignment along gaps, and (will have) confirmed these results through the subsequent closure of the gaps with PCR conducted by the software's designed primers.

Availability: This program is available free of charge for academic use. E-mail the author to receive a copy.

Contact: chaneylb@hiram.edu

Supplementary Materials: Primer3 for Windows is available upon request.

INTRODUCTION

The falling costs of extracting DNA sequence has resulted in an increasing number of institutions undergoing such research. As more and more locations become involved in these efforts, the average experience level of laboratory personnel undertaking this research drops substantially. With this in mind it has been recognized that the number, quality, accessibility, and ease-of-use of fully automated programs available for assisting at each step in a sequencing effort will greatly affect how monetarily and chronologically expensive any specific step in the process is. Analysis of the genome sequencing and preparation process reveals the following steps as computationally intensive: assembly, gap closure, finishing, and annotation. These specific distinctions are not as yet widely recognized in the literature and often little discrimination is made among tools which seek to minimize errors in the consensus sequence—the step here referred to as "finishing"—and those which seek to suggest the alignment of large regions of contiguous sequences generated by an assembler (contigs) relative to adjacent unsequenced regions (gaps)—here called "gap closure".

An analysis of the current software availability and ancestry for each field is shown through the entries located in Table 1. Due to time constraints, it would be impossible to conduct meaningful work in each of the fields; thus, selection of an individual field was required. Assembly was quickly discarded from consideration, given the duration (Dear and Staden, 1991) and range (Havlak *et al.*, 2004) of study on that problem. The proven usability of products such as Consed (Gordon *et al.*, 1998) and Autofinish (Gordon *et al.*, 2001) limited the degree of contribution which could be made to finishing, and this field was also discarded. The recent and rapid development of the two remaining fields, gap closure and annotation, made them appear equally well suited to development.

Gap closure was selected as the target field for a number of reasons. The high degree of interest and the slight degree of variation in approaches taken to

Table 1. Availability and history of software intended for automation of specific steps in the sequencing effort

Application		Publication
Assembly		
	xdap	Dear and Staden, 1991
	GAP	Bonfield *et al.*, 1995
	Atlas	Havlak *et al.*, 2004
Gap Closure		
	See Table 2	
Finishing		
	Consed	Gordon *et al.*, 1998
	Autofinish	Gordon *et al.*, 2001
Annotation		
	Imagene	Médigue *et al.*, 1999
	GeneQuiz	Andrade *et al.*, 1999
	Artemis	Rutherford *et al.*, 2000
	VISTA	Couronne *et al.*, 2003

Specific software products were selected on the basis of historical relevance, impact on other products in the field, or to demonstrate the most recent developments in the field.

the problem of automated decision making in gap closure, see Table 2, was one key element. Additionally, recent mathematical categorization (Wendl and Yang, 2004), of the degree to which further shotgun sequencing would fail to facilitate gap closure provided confirmation of previous observations “that certain regions of the genome…are very difficult to clone” (Herron-Olson *et al.*, 2003), highlighting the need for directed effort in gap closure. Personal confrontation with this problem in the sequencing effort of *Streptococcus sanguis* played a significant additional role in making this decision.

One approach offering substantial benefit would be to develop a cross-platform compatible application requiring minimum configuration and minimum technical background to use. The tool should seek integration of the maximum number of steps in the process of going from “assembly program output” to “ready-for-sequencing PCR product” and should allow customization of as many aspects as possible. Like previous work in the field, sequence comparison would be used as a tool for contig ordering. However, this application would generate ordered and directed pairs from the ends of two contigs not relying on any user determined individual related species, but rather sufficient homology with any known sequence.

RELEVANT WORK

Additional examination of the problem showed that varying levels of automation can be applied to gap closure. For example, some techniques have been developed which are automated but contain no components of decision making. One such example is multiplex PCR, a process which was initially developed to work with genes of known sequence (Burgart *et al.*, 1992) and recently modified to serve as a tool in genome closure (Tettelin *et al.*, 1999). Another such method is read pair identification, use of additional information stored by automatic sequencers about the origin of each read that in combination with assembly programs has been used to suggest contig alignment along gaps (Frohme *et al.*, 2001, Gordon *et al.*, 2001).

Some methods are also not highly suited to automation. One such method is the well known approach of physical mapping with restriction enzymes (Soulston *et al.*, 1988). Though still in use today (Weinel *et al.*, 2001), the method is time consuming, expensive, and may grow less useful as larger genomes are sequenced. Another older approach is that of PCR extension (Shymala and Ames, 1989) which remains in use today, with some modification (Carraro *et al.*, 2003). It is, however, so “straightforward” that little computational optimization can occur.

The bulk of recent development seeks to not only automate, to some extent, the gap closure process, but does so through methods that involve prediction of contig order and orientation. It should be noted that one characteristic shared by these tools, as seen in Table 2, is their reliance on the existence of some similar known sequence. This fact, coupled with the exponential growth in recent years of the number of known sequences, explains why they have only of late become significantly useful. This methodology of approaching the problem is in some ways an especially advantageous one, as it has long been observed that any tool implementing this strategy will only grow more helpful as the number of known sequences increases (Frangeul *et al.*, 1999).

The substantial limitations in the previously available tools can be classified into two categories. One group contains an inability to detect any pairs absent a single complete genome of a similar organism. The other class of tools requires similarity of a portion of the contig end, translated as a protein, to known protein sequence. A clear avenue for further research would be in optimizing searches for detecting meaningful hits in many different reference organisms, possibly occurring outside of protein coding regions. Additionally, the output of most of the programs requires some interpretation before primer design occurs, increasing the chance of human error.

ALGORITHM

Output of sequences from an assembly program is used as input to the **B**LAST **O**rganism **X**(cross) – **C**omparison (BOX-C) program. To increase compatibility specific formatting of the input file is not

Table 2. Analysis of software intended for automation of Gap Closure

Application Name	Notes on methodology Notes on output format(s)	Publication
GMPTB	Largest ORF in each contig end BLASTP comparison to protein database Text output only	Frangeul *et al.*, 1999
NUCmer	Whole contigs suffix tree aligned to single reference genome Text output only	Delcher *et al.*, 2002
PGAAS	Contig ends each undergo BLASTX comparison to protein database Graphical output in form of sequence alignment shown with actual sequence	Zhou *et al.*, 2002
MGView	Whole contigs BLASTN aligned to single reference genome Graphical output in form of PDF files	Herron-Olson *et al.*, 2003
Projector	Trimmed contig ends BLASTN aligned on template genome, followed by contig centers BLASTN aligned between ends or alignment discarded Graphical output in form of scalable vector graphics; primer design files	van Hijum *et al.*, 2003
CAAT-Box	Contig ends BLASTN or BLASTX aligned to single genome HTML text tables	Frangeul *et al.*, 2004

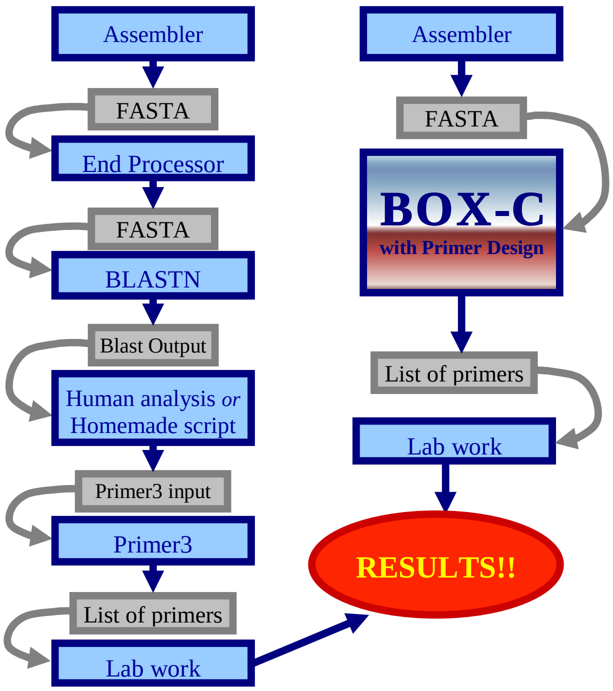

Fig. 1. The principle of BOX-C as demonstrated through comparison to the same method as performed before the program became available.

required, so long as each contig is given a unique identifier and some widely recognized standard file format is used in conjunction with an appropriate file extension. Preprocessing of the contigs occurs through the removal of a user specified excluded region of nucleotides from the extreme contig ends and the generation of a set of left and right ends of user defined size. The program uses a well recognized alignment algorithm, BLASTN (Altschul *et al.* 1990), in order to find regions of homology on the contig ends to known sequences. Each known sequence hit more than once by query sequences has the hits parsed. This parsing ensures that pairs of matches meeting certain compatibility and user specified range criteria are stored for subsequent primer design. Primer3 is used to accomplish the generation of oligonucleotides targeting each of the specific gaps between the suggested alignments. Production of the primers, conduction of the PCR reaction, and sequencing of the reaction product—all activities which cannot be completed or accurately simulated *in silico*—is all that stands in the way of closed gaps. Figure 1 provides a graphical comparison of the individual steps to those preformed by the program. As can be clearly seen, the program has successfully integrated a number of tedious steps and much file processing behind a single interface.

IMPLEMENTATION

The program, written in Perl, makes extensive use of the freely available BioPerl modules (http://www.bioperl.org/). The primary factors in this decision were the portability of Perl code from Windows to UNIX, as well as the independent maintenance of BioPerl code to help insure future compatibility. The program requires access to the NCBI remote BLAST server and a locally installed copy of Primer3.

Running remote BLAST through NCBI ensures that the database against which the search occurs is up to date. The inclusion of Primer3 is designed to guarantee the maximum possible degree of integration, thus making its porting necessary to ensure cross-platform operability. BOX-C is highly user configurable and is designed to be nearly self explanatory upon initial execution. Future improvements, such as the creation of a separate function for configuration, will allow users to modify the default values of many parameters, see Table 3, with a single command. This additional feature should also alleviate the need to manually enter the source file and encode the location of the Primer3 installation.

While BOX-C performs a similar function to the currently available programs, it excels in many areas. Previously mentioned is the high degree of configurability. This allows the user to determine the best settings for their particular organism and current degree of coverage, thus minimizing the number of missed or wrongly suggested gaps. As coverage increases and gaps decrease, users can relax restrictions on gap size to take into account large-scale rearrangement from known sequences; relax restrictions on e-values so that similar sequences outside of strong selection can be discovered; increase the allowed overlap, allowing BLAST to suggest possible contig alignments that may be missed because of misassembly upstream of the end; or perform almost any other optimization they have discovered.

BOX-C is also notably ahead of the field in its comparison of sequences at a nucleotide level to unrelated organisms. This technique has allowed local alignments of two contigs around a gap to be suggested in the *S. sanguis* genome by hits to sequences as distantly related as those from *Homo sapiens*. Unlike most of its competition, it also attempts to go beyond identification of contig pairs, taking the additional measure of suggesting primers that might be used in conjunction with PCR to cross the putative gap. This alleviates some possibility that the user might incorrectly interpret the results, and in many cases will even suggest primers that will attempt to get sequence from the direct and complementary strands.

The program allows intermediate files to be output at many stages in the process, allowing the user to observe, at any convenient time, what data has been used, and grants the opportunity to cross verify the results the program generates. However, none of the output is mandatory, avoiding the waste of space which occurs with the generation of unnecessary files.

Table 3. A list of BOX-C parameters with explanation of the function of each

Parameter identifier	Default value (if present)	Type of argument expected	Description of function
-h		None	If present, a help message is displayed
-v	*true*	"0" or none	Verbose output of debugging information
-i[1]		Path[2]/filename	Location of the contig file; *optional*- may be included a second time with location of intermediate BLAST results file
-f	ps	ebmps	Outputs files corresponding to each given letter: e: ends file generated in preprocessing b: BLAST results as individual files m: matches, or paired contigs, found by the program p: primer pairs for each match s: summary of where the matches were found
-o	box-c.fasta	Path[2]/filename	Basic filename given to output files, will be slightly modified for output of each specific file
-r	*false*	None	If passed as an unvalued argument, ends generated by contigs where the ends are not entirely unique will not be considered
-e	500	Integer[3]	Size in base pairs of ends taken for comparison
-y	0	Integer[3]	Size in base pairs of contig end-regions excluded before ends of the specified length are taken
-l	0	Integer[3]	Number of base pairs which can overlap on a BLAST-matched sequence before match is excluded as something which should have been identified by the assembler
-g	3000	Integer[3]	Maximum number of base pairs which separate two possible contig ends in order for ends to be considered a likely pair
-t	1e-20	BLAST-style e-val	Threshold of BLAST hits to consider for alignment
-d	nr	BLAST database	BLAST database against which to search (see http://www.ncbi.nlm.nih.gov/BLAST/ for a list)
-w	15	Integer	Time in seconds to wait between requests to NCBI BLAST server (minimum of 5 seconds)

[1] -must be given and be the location of the file of contigs to be processed

[2] -optional

[3] -must be non-negative

DISCUSSION

The demonstrated ability of BOX-C to suggest meaningful alignments (Kitten and Chaney, unpublished data) helps categorize the program as a success. Despite some issues with configuration of BioPerl and Primer3, the program's general ease of use and cross-platform portability should greatly increase the accessibility to groups in need of such a product.

The program has met all of the initial goals of the project to varying extents. Perhaps most satisfying is the ease of use and masking of details one experiences when using the current version of the software. These features demonstrate the successful development in the most important categories.

One category is fallen somewhat short in, as the simplest format of a matched pair currently output is the summary of the results, which by name indicate the order and orientation of the contigs involved in the pair. The greatest disappointment has been this failure to create a "simple, visual" style of output for understanding the suggested relationships between contigs. In the same vein, the absence of a graphical user interface may prevent use of the program by those who would potentially benefit. Future expansion to the program will likely be directed in these areas first.

Should the program be re-developed, it is likely implementation decisions would be made sooner so that actual work could be underway by no later than the third week. This has further emphasized the lesson that, despite the importance of development decisions, they should be completed relatively early in the software development timeline.

After the benefits of the project were examined, one such crucial decision, the choice of Perl as the language of implementation, has proven highly educational. The development of this program has provided an invaluable degree of further understanding of the language. Within the code itself there are demonstrable examples of how the features provided by the language were more fully exploited towards the conclusion of the venture. This increased comprehension has indicated that the use of subroutines would improve the readability and flexibility of the code. This is an additional area that would be differently implemented given the chance and will likely be improved in future development.

The project provided a long list of satisfactory results. Foremost among these are a better understanding of the Perl language, attaining familiarity with the BioPerl modules, and more fully understanding the process of gap closure. It is felt that the finished program adequately demonstrates these traits while holding its own as a product currently competitive with the most advanced software available in the field.

REFERENCES

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) A basic local alignment search tool. *J. Mol. Biol.*, **215**, 403-410.

Andrade,M.A., Brown,N.P., Leroy,C., Hoersh,S., Daruvar,A., Reich,C., Franchini,A., Tamames,J., Valencia,A., Ouzounis,C. and Sander,C. (1999) Automated genome sequence analysis and annotation. *Bioinformatics*, **15**, 391-412.

Bonfield,J.K., Smith,K.F. and Staden,R. (1995) A new DNA sequence assembly program. *Nucleic Acids Res.*, **23**, 4992-4999.

Burgart,L.J., Robinson,R.A., Heller,M.J., Wilke,W.W., Iakoubova,O.K. and Cheville,J.C. (1992) Multiplex polymerase chain reaction. *Mod. Pathol.*, **5**, 320-323.

Carraro,D.M., Camargo,A.A., Salim,A.C., Grivet,M., Vasconcelos,A.T. and Simpson,A.J. (2003) PCR-assisted contig extension: stepwise strategy for bacterial genome closure. *Biotechniques*, **34**, 626-632.

Couronne,O., Poliakov,A., Bray,N., Ishkhanov,T., Ryaboy,D., Rubin,E., Pachter,L. and Dubchak,I. (2003) Strategies and tools for whole-genome alignments. *Genome Res.*, **13**, 73-80.

Dear,S. and Staden,R. (1991) A sequence assembly and editing program for efficient management of large projects. *Nucleic Acids Res.*, **19**, 3907-3911.

Delcher, A. L., Phillippy, A., Carlton, J. and Salzberg, S. L. (2002) Fast algorithms for large-scale genome alignment and comparison. *Nucleic Acids Res.*, **30**, 2478-2483.

Frangeul,L., Glaser,P., Rusniok,C., Buchrieser,C., Duchaud,E., Dehoux,P. and Kunst,F. (2004) CAAT-Box, contigs-assembly and annotation tool-box for genome sequencing projects. *Bioinformatics*, **20**, 790-797.

Frangeul,L., Nelson,K.E., Buchrieser,C., Danchin,A., Glaser,P. and Kunst,F. (1999) Cloning and assembly strategies in microbial genome projects. *Microbiology*, **145**, 2625-2634.

Frohme,M., Camargo,A.A., Czink,C., Matsukuma,A.Y., Simpson, A.J.G., Hoheisel,J.D. and Verjovski-Almedia,S. (2001) Directed gap closure in large-scale sequencing projects. *Genome Res.*, **11**, 901-903.

Gordon,D., Abajian,C. and Green,P. (1998) Consed: a graphical tool for sequence finishing. *Genome Res.*, **8**, 195-202.

Gordon,D., Desmarais,C. and Green,P. (2001) Automated finishing with Autofinish. *Genome Res.*, **11**, 614-625.

Havlak,P., Chen,R., Durbin,J., Egan,A., Ren,Y. Song,X., Weinstock,G.M. and Gibbs,R.A. (2004) The Atlas genome assembly system. *Genome Res.*, **14**, 721-732.

Herron-Olson,L., Freeman,J., Zhang,Q., Retzel,E.F. and Kapur,V. (2003) MGView: an alignment and visualization tool to enhance gap closure of microbial genomes. *Nucleic Acids Res.*, **31**, e106.

Médigue,C., Reschenmann,F., Danchin,A. and Virai,A. (1999) Imagene: an integrated computer environment for sequence annotation and analysis. *Bioinformatics*, **15**, 2-15.

Rutherford,K., Parkhill,J., Crook,J., Horsnell,T., Rice,P., Rajandream,M. and Barrell,B. (2000) Artemis: sequence visualization and annotation. *Bioinformatics*, **16**, 944-945.

Shymala,V. and Ames,G.F. (1989) Genome walking by single-specific-primer polymerase chain reaction: SSP-PCR. *Gene*. **84**, 1-8.

Sulston,J., Mallett,F., Staden,R., Durbin,R., Horsnell,T. and Coulson,A. (1988) Software for genome mapping by fingerprinting techniques. *Comput. Appl. Biosci.*, **4**, 125-132.

Tettelin,H., Radune,D., Kasif,S., Khouri,H. and Salzberg,S.L. (1999) Optimized multiplex PCR: efficiently closing a whole-genome shotgun sequencing project. *Genomics*, **62**, 500-507.

van Hijum,S.A.F.T., Zomer,A.L., Kuipers,O.P. and Kok,J. (2003) Projector: automatic contig mapping for gap closure purposes. *Nucleic Acids Res.*, **31**, e144.

Weinel,C., Tummler,B., Hilbert,H., Nelson,K.E. and Kiewitz,C. (2001) General method of rapid Smith/Birnstiel mapping adds for gap closure in shotgun microbial genome sequencing projects: application to *Pseudomonas putida* KT2440. *Nucleic Acids Res.*, **29**, e110.

Wendl,M.C. and Yang,S.P. (2004) Gap statistics for whole genome shotgun DNA sequencing projects. *Bioinformatics*, submitted manuscript.

Zhou,Y. Li,T., Zhao,J. and Luo,J. (2002) PGAAS: a prokaryotic genome assembly assistant system. *Bioinformatics*, **18**, 661-665.