Management and visualization of whole genome shotgun assemblies using SAM

René L. Warren, Yaron S. Butterfield, Ryan D. Morin, Asim S. Siddiqui, Marco A. Marra, and Steven J.M. Jones
Genome Sciences Centre, Vancouver, BC, Canada

BioTechniques 38:715-720 (May 2005)

We have designed and implemented a system to manage whole genome shotgun sequences and whole genome sequence assembly data flow. The Sequence Assembly Manager (SAM) consists primarily of a MySQL relational database and Perl applications designed to easily manipulate and coordinate the analysis of sequence information and to view and report genome assembly progress through its Common Gateway Interface (CGI) web interface. The application includes a tool to compare sequence assemblies to fingerprint maps that has been used successfully to improve and validate both maps and sequence assemblies of the Rhodococcus *sp. RHA1 and* Cryptococcus *neoformans WM276 genomes.*

INTRODUCTION

Despite recent advancements in computer technology, the amount and rate of the data generated by high-throughput DNA sequencing centers have brought new challenges and constraints to the development of effective sequence assembly and data management tools. Among the first such efforts was that of Lawrence and co-workers (1) 10 years ago, with the development of the Genome Reconstruction Manager (GRM), a complete software package to assemble and view genomic sequences. Although restricted to its own assembly algorithms, the system employed an object database, allowing efficient data storage and multi-user access. Since the GRM, numerous proprietary tools have been developed to perform small-scale sequence assemblies and contig visualization on personal computers. The complexity of genomes and software availability have limited the general use of these tools. In response to biological challenges in the field of genomics, several improved fragment assembly programs such as the Celera Assembler (2), Arachne (3), Phusion (4), and PCAP (5) have been developed. Unfortunately, the diversity in the output of whole genome shotgun (WGS) assemblers is not always amenable to genome visualization software and mainstream finishing tools such as gap4 (6) and Consed (7).

We have developed a system to render whole genome assembly (WGA) data generic, upon which genome analysis tools and viewers for different sequence assemblers were designed. The key component of this application is a MySQL relational database (SAMdb), a central repository for storing sequencing data, assembly information, fingerprint map data, genome analysis data, gene predictions, and annotation information. The main purpose of the Sequence Assembly Manager (SAM) is to organize WGA in a structured way such that users can assess WGA quality rapidly using an intuitive web interface that allows easy evaluation of the assembly quality and its progress.

METHODS AND ALGORITHMS

SAM is implemented in Perl and runs on Linux. It is distributed under the same term as Perl and MySQL and is available from the Canada's Michael Smith Genome Sciences Center (Vancouver, BC, Canada) at www.bcgsc.ca/bioinfo/software/sam. SAM is free for academic and noncommercial use. The user may modify the software but must make the changes available to its creators. Any modifications are subject to and must include a copy of our license, available at www.bcgsc.ca/bioinfo/software/gsc-software-license. A minimum of 7 MB of disk space is required to install SAM.

SAMdb and the System's Input

SAM requires a single input file: a collection of sequence reads and associated ancillary information in an XML file format. The XML format definition is largely derived from the National Center for Biotechnology Information (NCBI; Bethesda, MD, USA) Trace Archive (www.ncbi.nlm.nih.gov/Traces/trace.cgi). Optional information about clone fingerprint maps can also be added to the system by means of an XML file. Having the sequence read, clone, and library information in a single relational database gives enormous flexibility to WGS projects; a subset of sequences meeting specific criteria can be quickly chosen and assembled using this system. SAMdb stores the results of WGA, creating a relationship between every assembled read, contig, supercontig, and ultracontig. Genome analysis induces further ramifications of these relationships, creating an organized web of information between the raw sequence data and the annotated genome.

System

SAM is written in Perl, and the scripts and libraries fall into four main categories: (*i*) system utilities; (*ii*) custom assembly and gene prediction; (*iii*) WGA analysis; and (*iv*) user interface and web utilities. System utility programs and libraries include XML parsers and database connection and interaction modules. Wrapper modules and shell scripts written for specific assembly programs coordinate the assembly and the information retrieval. Similarly, the execution of gene prediction programs is controlled by wrappers written specifically for each application. SAM currently supports the Phrap (www.phrap.org) and Arachne (3) assemblers as well as the Glimmer (8) and GeneMarkS (9) gene prediction programs. The sequence assemblers were compiled on an AMD® 64-bit Opteron™ computer

(AMD, Sunnyvale, CA, USA) running SUSE 9.0. Phrap and Arachne required less than 4 GB of random access memory (RAM) to run on genomes comprising up to 19 million bp. Software settings, configuration, and run parameters are stored in SAMdb and adjusted to reflect the needs of the project. In-depth analysis of WGA information such as the evaluation of sequence gap size, orientation of contigs, identification of gap-spanning and overlapping clones, calculation of clone insert size distribution, and library-specific read distribution within the assembly is automated by a series of modules that analyze the assembly once it is in SAMdb. Functions within these modules are executed whenever a sequence assembly is launched or imported into the system.

User Interface

WGA data stored in SAMdb is accessed by simply running the Perl CGI application sam.pl in a web browser. The user interface is designed for both project collaborators and administrators. From the web-based interface, project administrators can additionally perform custom assemblies, import sequence reads and maps, link fingerprint information to existing sequence assemblies, and predict genes, without prior knowledge of the gene prediction or assembly software. These tasks can be run at the UNIX prompt or executed from the web interface. With the latter, jobs are placed in a queue (SAMdb, Job_Queue table), and a cronjob running periodically executes the tasks serially on computers specified in the Node table.

Features of the user interface include a page to allow data comparison between different assembly builds obtained using the same or different WGS assemblers. The data being compared range from simple WGA statistics to calculation estimates based on the assembly data. The page shows relevant data for each build, along with the change, if applicable. Dynamically generated graphs showing the assembly progress (Figure 1), contig relationships, sequence gaps, clone insert sizes, and read distribution are designed to assist project investigators in monitoring the quality of both the sequence assembly and genomic libraries as the shotgun depth increases.

Genomic views of the assembly are available using the Contig View page. This viewer allows precomputed automated gene predictions and annotations (based on sequence similarity) to be displayed on both strands for every contig of any given assembly. This feature is not intended to replace more advanced genome browsers, but rather provides a preliminary assessment of the gene composition and confidence in a convenient package. Base-quality thresholds are stored in SAMdb and used to filter putative genes meeting strict quality requirements, adding strength to the predictions and downstream experiment design.

Relationships between sequence assembly supercontigs and restriction fragment clone fingerprint map contigs created by the FPC package (10,11) are viewed by accessing the Map View feature. Provided that fingerprinted bacterial artificial chromosome (BAC)/fosmid clones have their ends sequenced and are incorporated into the assembly, a supercontig layout and orientation is computed for every fingerprint map contig. Fingerprint map information, contig-supercontig relationships, and sequence read positions are first extracted from the database. If a sequence assembler does not compute scaffolds, the algorithm assumes that every sequence contig is also a supercontig. Read coordinates are converted to their relative position on their supercontig. For every fingerprint map contig and each of the fingerprinted clones comprising it, we record whether the mate pairs for that clone have been assembled into the same supercontig. If not, we have identified a clone joining and orientating two supercontigs. Such ultracontigs are built based on (*i*) fully assembled clones and (*ii*) the presence of at least two mate pairs spanning a gap between

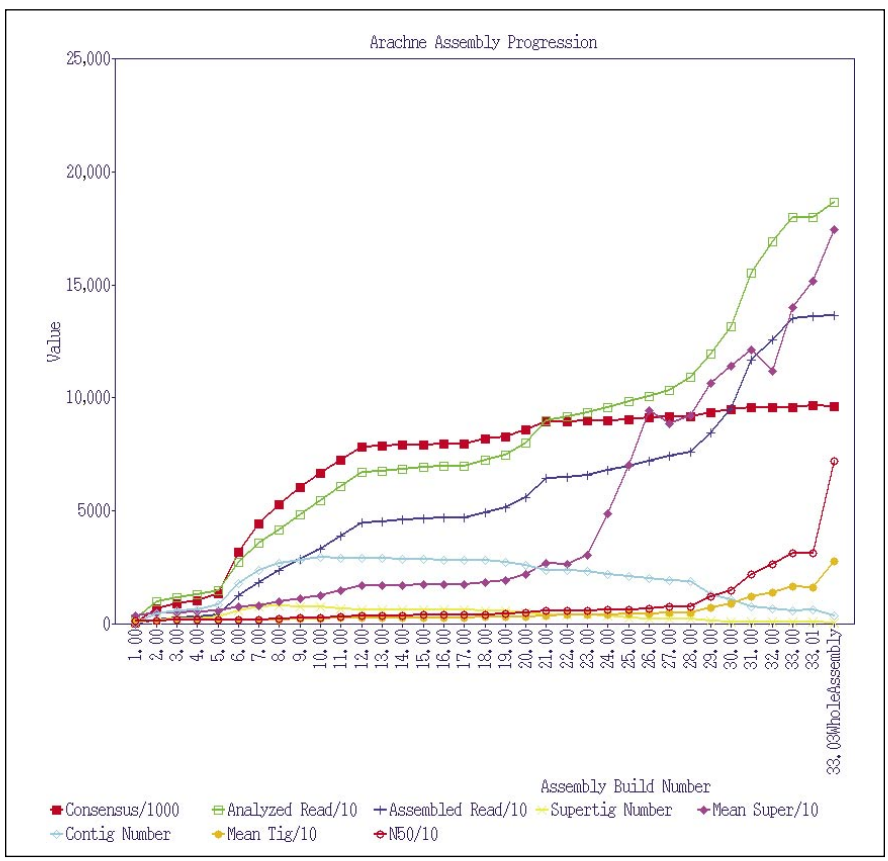

Figure 1. Sample output from sam.pl. The graph shows the Arachne assembly progression of user-selected markers for the *Rhodococcus* sp. RHA1 genome. This graph provided the first insights on the genome size for this organism.

two supercontigs. Once the supercontig layout has been determined, the order and orientation are computed, based on fingerprinted clone overlaps and mate pair direction, respectively. The resulting ultracontig enables the user to gain further insight into the higher organization of WGA and identify all the supercontigs that have a relationship with the fingerprint map. This tool also provides an effective means of high-throughput quality control by visualizing the clone tiling path relative to the sequence assembly and by displaying in silico digestion band patterns predicted from DNA sequences and comparing them to the experimental fingerprint for each clone of a map contig that projects onto the sequence assembly (Figure 2).

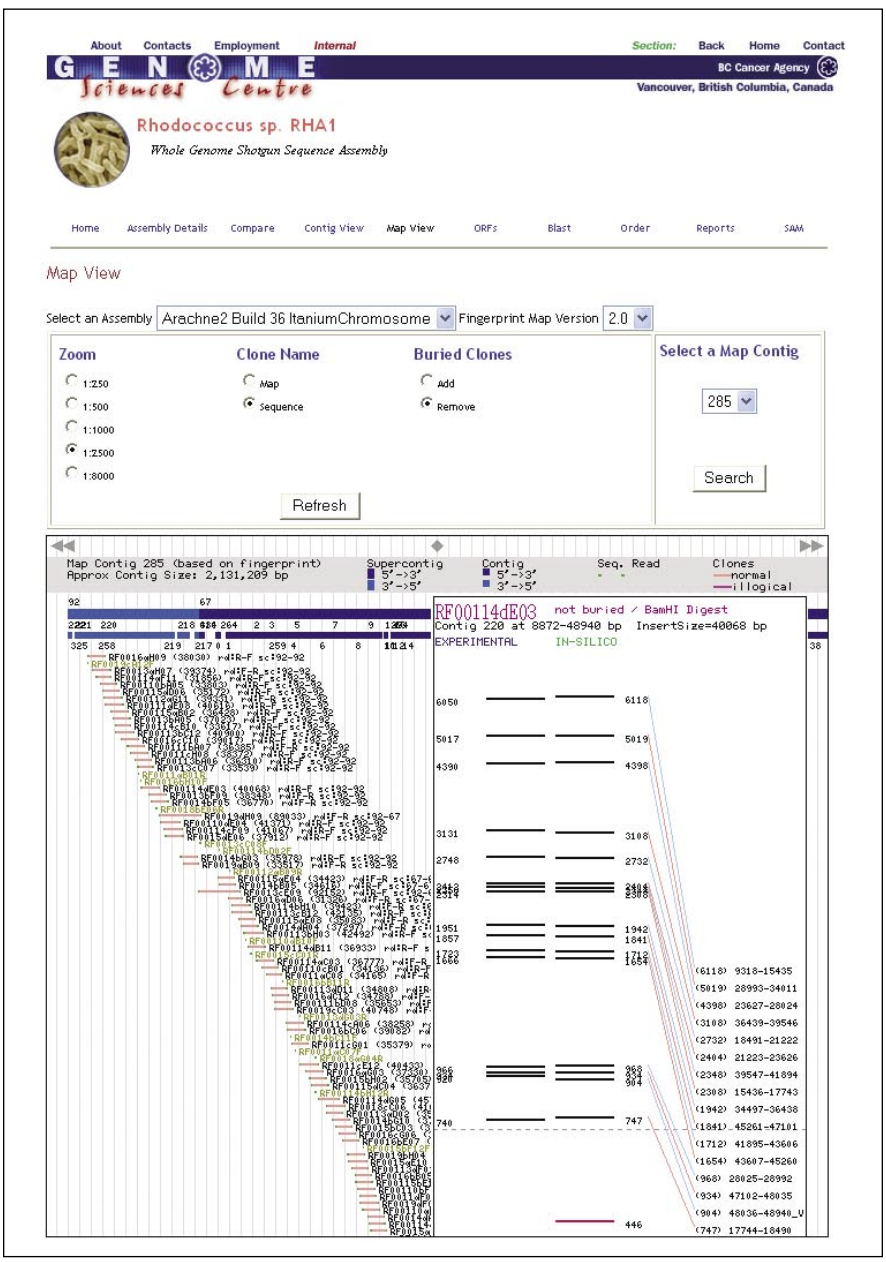

Figure 2. Image generated dynamically using the Map View feature of sam.pl. The green dots show the position of fingerprinted clone end-reads in the sequence assembly for every clone overlapping in a given map contig. The joining line indicates the status of the clone, based on the orientation of the reads and their distance apart from each other. Both the position of the fingerprinted clones and in silico digests are used to validate the sequence assembly. Ultracontigs are displayed if at least two clones span the same two sequence supercontigs.

RESULTS AND DISCUSSION

We have used SAM to assemble, analyze, and annotate the genome of *Rhodococcus* sp. RHA1, a polychlorinated biphenyl (PCB)-degrading bacterium (12). The Map View feature was used recently to validate the sequence of this genome and improve the contiguity of the corresponding fingerprint map by facilitating the identification of sequence supercontigs containing clones from two or more map contigs. In silico digests of precise genomic regions along with BAC/fosmid clone localization within these supercontigs have allowed us to merge and incidentally decrease the number of map contigs from 60 to just 5 for this organism. Furthermore, manual sequence finishing of the *Rhodococcus* sp. RHA1 genome was achieved using SAM. Our main finishing strategy involved aligning the edges of contigs flanking a gap against contigs in another low-stringency assembly stored in SAMdb. The matching region of the hit contig comprised a subset of reads that were extracted from the database. These reads were reassembled and used to close the gap. At its peak throughput, this technique allowed the closure of 70 sequence gaps in a 2-week period.

The efficiency of this system relies on two pillars: (*i*) a MySQL relational database that standardizes the data obtained from different WGA, gene predictions, and fingerprint maps; and (*ii*) a comprehensive web interface intended to vulgarize sequence assemblies, promote discoveries, and facilitate downstream research. The web interface displays a wealth of information on the sequence assembly and genome annotation, including a complete characterization of all sequence gaps and read overlaps and clone, contig, and supercontig size distribution. It generates custom reports showing side-by-side comparisons of assemblies obtained with the same or different sequence assemblers. It also reports the assembly progress by displaying assembly-specific markers. A complete list of open reading frames (ORFs) along with their best BLAST (Basic Local Alignment Search Tool; Reference 13) alignments to public sequences is reported in both table

form and graphically, by showing the ORFs in the context of its sequence contig for any given assembly. SAM allows users to integrate information from genome maps and validate the sequence assembly using user-friendly graphics and intuitive navigation tools. The user interface has a dynamic component where users can assemble selected reads, run gene prediction programs, or perform sequence alignments to genome assemblies using a BLAST page, without prior knowledge of the underlying algorithms and their run options.

SAM was used to manage the genome assemblies and analyses of the PCB-degrading bacterium *Rhodococcus* sp. strain RHA1 and the fungal pathogen *Cryptococcus* neoformans strain WM276. It was used successfully to: (*i*) improve and validate both fingerprint maps and sequence assemblies; (*ii*) identify problematic genomic libraries, monitor the assembly progress, compare sequence assemblers, and assess the assembly quality; (*iii*) predict and annotate genes for use in probe design for microarray experiments; (*iv*) develop strategies for sequence assemblies and finishing; and (*v*) automate clone ordering for gene knockout experiments. SAM will prove useful to other groups interested in managing multiple WGA simultaneously, developing tools that require sequence assembly data in a generic format, and designing efficient strategies for sequence finishing.

ACKNOWLEDGMENTS

We thank Jerry Liu and Kevin Teague from the Genome Sciences Centre for helpful suggestions and discussions on MySQL and web development. This project was supportd by a grant from Genome British Columbia and Genome Canada. M.A.M. and S.J.M.J. are Michael Smith Foundation for Health Research Scholars.

COMPETING INTERESTS STATEMENT

The authors declare no competing interests.

REFERENCES

1. **Lawrence, C.B., S. Honda, N.W. Parrott, T.C. Flood, L. Gu, L. Zhang, M. Jain, S. Larson, et al.** 1994. The Genome Reconstruction Manager: a software environment for supporting high-throughput DNA sequencing. Genomics *23*:192-201.
2. **Huson, D.H., K. Reinert, S.A. Kravitz, K.A. Remington, A.L. Delcher, I.M. Dew, M. Flanigan, A.L. Halpern, et al.** 2001. Design of a compartmentalized shotgun assembler for the human genome. Bioinformatics *17*:(*Suppl 1*)S132-S139.
3. **Batzoglou, S., D.B. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J.P. Mesirov, et al.** 2002. ARACHNE: a whole-genome shotgun assembler. Genome Res. *12*:177-189.
4. **Mullikin, J.C. and Z. Ning.** 2003. The phusion assembler. Genome Res. *13*:81-90.
5. **Huang, X., J. Wang, S. Aluru, S.-P. Yang, and L. Hillier.** 2003. PCAP: a whole-genome assembly program. Genome Res. *13*:2164-2170.
6. **Dear, S. and R. Staden.** 1991. A sequence assembly and editing program for efficient management of large projects. Nucleic Acids Res. *19*:3907-3911.
7. **Gordon, D., C. Abajian, and P. Green.** 1998. Consed: a graphical tool for sequence finishing. Genome Res. *8*:195-202.
8. **Delcher, A.L., D. Harmon, S. Kasif, O. White, and S.L. Salzberg.** 1999. Improved microbial gene identification with Glimmer. Nucleic Acids Res. *27*:4636-4641.
9. **Besemer, J., A. Lomsadze, and M. Borodovsky.** 2001. GeneMarkS: a self-training method for gene starts in microbial genomes. Implication for finding sequence motifs in regulatory regions. Nucleic Acids Res. *29*:2607-2618.
10. **Ness, S.R., W. Terpstra, M. Krzywinski, M.A. Marra, and S.J. Jones.** 2002. Assembly of fingerprint contigs: parallelized FPC. Bioinformatics *18*:484-485.
11. **Soderlund, C., I. Longden, and R. Mott.** 1997. FPC: a system for building contigs from restriction fingerprinted clones. Comput. Appl. Biosci. *13*:523-535.
12. **Warren, R., W.W.L. Hsiao, H. Kudo, M. Myhre, M. Dosanjh, A. Petrescu, H. Kobayashi, S. Shimizu, et al.** 2004. Functional characterization of a catabolic plasmid from polychlorinated-biphenyl-degrading *Rhodococcus* sp. strain RHA1. J. Bacteriol. *186*:7783-7795.
13. **Altschul, S.F., W. Gish, W. Miller, E.W. Myers, and D.J. Lipman.** 1990. Basic local alignment search tool. J. Mol. Biol. *215*:403-410.

Received 27 October 2004; accepted 30 November 2004.

Address correspondence to Steven J.M. Jones, Canada's Michael Smith Genome Sciences Centre, Vancouver, BC V5Z 4S6, Canada e-mail: sjones@bcgsc.ca