CloVR-Microbe: Assembly, gene finding and functional annotation of raw sequence data from single microbial genome projects – standard operating procedure, version 1.0

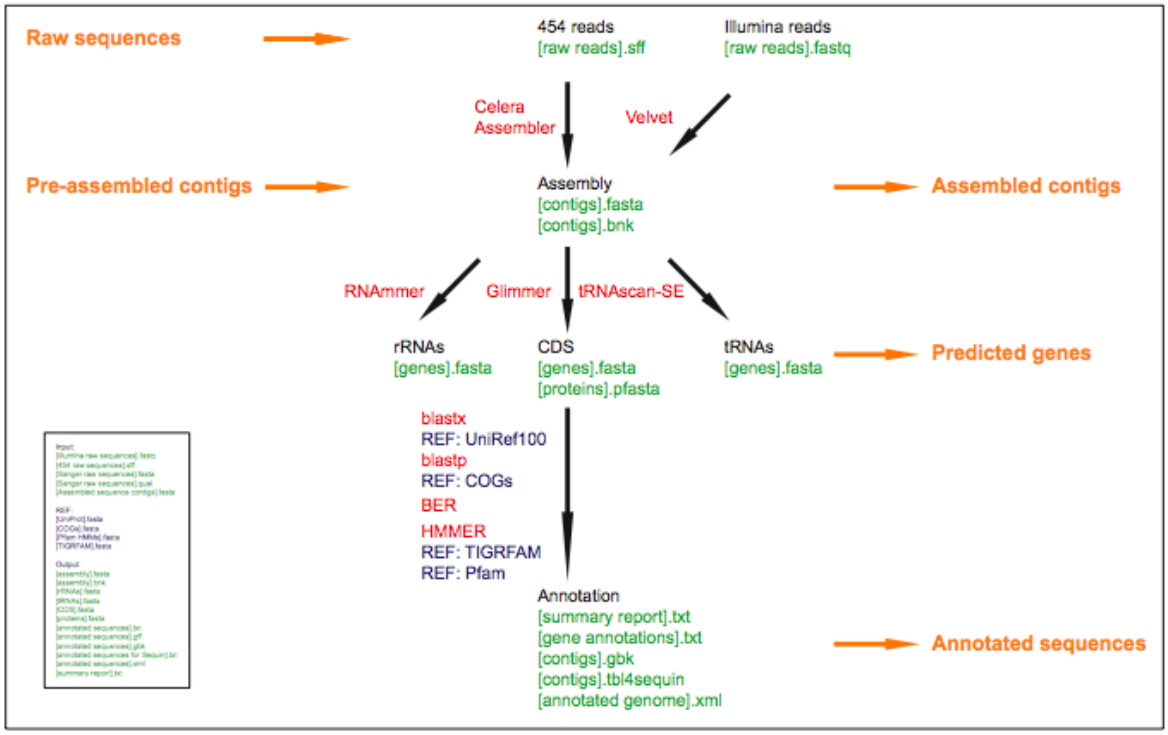
Kevin Galens, James Robert White, Cesar Arze, Malcolm Matalka, Michelle Gwinn Giglio, the CloVR team, Samuel V. Angiuoli & W. Florian Fricke

The Institute for Genome Sciences, University of Maryland School of Medicine, Baltimore, MD 21201, USA

Abstract

The CloVR-Microbe pipeline performs the basic processing and analysis steps required for standard microbial single-genome sequencing projects: A) Whole-genome shotgun sequence assembly; B) Identification of protein and RNA-coding genes; and C) Functional gene annotation. B) and C) are based on the IGS Annotation Engine (http://ae.igs.umaryland.edu/), which is described elsewhere (K Galens et al. *submitted*). The assembly component of CloVR-Microbe can be executed independently from the gene identification and annotation components. Alternatively, pre-assembled sequence contigs can be used to perform gene identifications and annotations. The pipeline input may consist of unassembled raw sequence reads from the Sanger, Roche/454 GS FLX or Illumina GAII or HiSeq sequencing platforms or of combinations of Sanger and Roche/454 sequence data. The pipeline output consists of results and summary files generated during the different pipeline steps. Annotated sequence files are generated that are compatible with common genome browser tools and can be submitted to the GenBank repository at NCBI. This protocol is available in CloVR beta versions 0.5 and 0.6.

Overview



Software

Step	Program	Version	Weblink	Reference
Assembly	Celera Assembler (CA)	5.4	http://sourceforge.net/apps/mediawiki/wgs-assembler/index.php?title=Main_Page	[1]
	Velvet	0.7.55	http://www.ebi.ac.uk/~zerbino/velvet/	[2]
Gene Finding	Glimmer	3.02	http://www.cbcb.umd.edu/software/glimmer/	[3]
	tRNAscan-SE	1.23	http://www.bioinformatics.org/wiki/TRNAscan-SE	[4]
	RNAmmer	1.2	http://www.cbs.dtu.dk/services/RNAmmer/	[5]
Annotation	Emboss	5.0	http://emboss.sourceforge.net/	[6]
	BLAST	2.2.21	http://blast.ncbi.nlm.nih.gov/Blast.cgi	[7]
	HMMer	2.3.2	http://hmmer.janelia.org/	[8]

Reference data

Database	Data	Version	Weblink	Reference
TIGRFAM	Protein families	7.0	http://www.jcvi.org/cms/research/projects/tigrfams/	[9]
Pfam	Protein families	22.0	http://pfam.sanger.ac.uk/	[9]
COG	Clusters of orthologous genes	1.0	http://www.ncbi.nlm.nih.gov/COG/	[10]
UniRef100	Proteins	Oct. 2010	http://www.ebi.ac.uk/uniref/	[11]

Pipeline input

Data	Suffix	Description
Raw sequence reads	.sff	454 sequencer output
Raw sequence reads	.fastq	Illumina sequencer output
Sequence reads	.fasta	Multiple sequence file
Sequence read qualities	.qual	Quality values corresponding to multiple sequence file
Pre-assembled contigs	.fasta	Multiple sequence file

Pipeline output

Data	Suffix	Description
Assembled contigs	.fasta	Multiple sequence file
	.bnk	Celera sequence assembly, compatible with Hawkeye [12] assembly viewer
Predicted genes	.coords	Coordinates table
	.fasta	Genes sequences (nucleotides)
	.pfasta	Protein sequences (amino acids)
Annotated sequences	.txt	Gene coordinates and annotation table
	.gbk	GenBank sequence file, compatible with Artemis [13] genome browser
	.tbl4sequin	Feature table for NCBI Sequin submission tool
	.xml	Machine-readable annotation file
Summary report	.txt	Pipeline summary report

A. Assembly

During the assembly process the fragmentary whole-genome shotgun (WGS) sequence data typically produced in microbial single-genome projects are used to reconstruct long contiguous sequences or *contigs*. Scaffolds are composed of multiple contigs that are linked by paired-end reads. In the assembly output each scaffold will be represented by a single sequence (FASTA) in which stretches of "N"s indicate gaps and estimated gap lengths spanned by paired-end reads.

A.1. Procedure

A.1.1. Assembly with the Celera Assembler

A.1.1.1 Raw sequence data file conversion

The Celera Assembler [1] uses .frg files as input, which are generated from 454 or Sanger raw sequences or from combinations of both data types with the sffToCA tool. Raw sequences from the 454 pyrosequencing platform are accepted either as single read or mated paired-end reads in the .sff format. Raw sequences from the Sanger platform are accepted as pairs of .fasta and .qual files with the same name prefix. As a default, sffToCA is run with the following parameters, which were optimized for data generated with the 454 GS FLX XLR Titanium platform: "-clear 454" sets the clear range for each sequence read "as is", using the clear range determined by the 454 sequencing machine; "-trim chop" erases sequences outside of the 454 clear ranges. If paired-end data are being used as input, a 454 linker has to be specified as either "-linker flx" or "-linker titanium", depending on the 454 sequencing platform generation that is being used. In addition, an insert size range has to be selected, e.g. "8000 1000" where the first number specifies the average insert length (8 kbp) and the second number the standard deviation (1 kbp). More than one .sff file can be used as input for sffToCA, which produces a single .frg file that serves as input for the assembly step.

A.1.1.2. Assembly

For the assembly process, Celera Assembler is run with the "runCA" executive script, using default parameters. Alternatively, a "spec file" can be provided by the user to select alternative parameters for the assembly process. Spec file examples can be downloaded from the Celera Assembler documentation page (http://sourceforge.net/apps/mediawiki/wgs-assembler/index.php?title=RunCA_ Examples). The assembly step generates a number of different output files. First, assembly statistics are collected in a .txt summary report file. Second, .fsa (FASTA) files are generated as the direct assembly output, which serve as the input for the Gene Finding step. Third, a .asm file is generated, which can be used in combination with the .frg file from the sffToCA output to generate the .bnk file for visualization of the assembly results with the Hawkeye assembly viewer [12].

A.1.1.3. Celera assembly visualization

Using information from the raw sequence data (.frg) and the sequence assembly (.asm), a .bnk file is generated with the "toAmos" and "bank-transact" tools (http://sourceforge.net/apps/mediawiki/amos/index.php?title=Hawkeye). The .bnk file serves as the direct input for the Hawkeye assembly viewer [12], which itself is not part of the CloVR-Microbe pipeline and which is not installed on the CloVR virtual machine image.

A.1.2. Assembly with the Velvet Assembler

In the case of assembling Illumina-based WGS sequence data, the Velvet assembly program [2] can accept as input any combination of short or long read data (with or without paired ends) in FASTQ or FASTA format. Additionally, start and end hash length parameters are required for kmer counting that Velvet performs, which must be odd values, and by default are "19" (start) and "31" (end). For paired-end data an insert-size and standard deviation must be provided, e.g.

"-ins_length 300 -ins_length_sd 50". Velvet outputs a .fasta file of all sequence contigs of at least 500 bp in length.

A.1.3. Split assembly into separate contigs
The following two steps of the CloVR-Microbe pipeline, B) Gene Finding and C) Annotation, are performed on each individual scaffold and/or contig from the WGS assembly separately. This is advantageous, as it allows for parallelization of processes.

B. Gene Finding
B.1. Identification of RNA genes
B.1.1. Identification of ribosomal RNA genes
Ribosomal RNA (rRNA) genes are identified with the RNAmmer tool [5]. Contig FASTA files (.fsa) are used as input to run RNAmmer with the "bac" option, which specifies "Bacteria" as the superkingdom of the input sequence source and the "lsu,tsu,ssu" option, which specifies the molecule types to search for as 5S/8S rRNA, 16S/18S rRNA, and 23S/28S rRNA. This step generates as output a .txt summary report file and gene coordinate files in the BSML, .xml, and .gff file formats, though these particular files are not downloaded at the end of the pipeline.

B.1.2. Identification of transfer RNA genes
Transfer RNA (tRNA) genes are identified with the tRNAscan-SE tool [4], which is run on FASTA files (.fsa) with the "-B" option to use the bacterial model for the prediction of tRNA genes. This step generates .txt coordinate file of all tRNA genes and a BSML (.bsml) file as output.

B.2. Identification of coding genes (CDS)
Open reading frames (ORFs) coding for proteins (CDS) are identified using the Glimmer3 software package [3]. The execution procedure is derived from the iterative self-training mode of operation described in the software documentation, which is available from the project website (http://www.cbcb.umd.edu/software/glimmer).

B.2.1. Identification of training set to build gene prediction model
The "long-orfs" program is run to identify long, non-overlapping ORFs from the total genome assembly, including all sequence scaffolds and/or contigs. These long ORFs serve as a training set to build the interpolated context model (ICM), which is used to predict CDS. The long-orfs program is run with the "-n" or "--no_header" option and the "-t 1.15" option, which specifies the cutoff for the entropy distance score used to select the training set of ORFs. The output is a list of non-overlapping ORFs.

B.2.2 Generation of interpolated context model
An interpolated context model (ICM) is built with data from the long-orfs output using the "build-icm" program with default options.

B.2.3 Glimmer3 gene prediction
B.2.3.1. First iteration
The first iteration of gene finding is run with the ICM model from B.2.2 using the "glimmer3" program executed with the following parameters: "-o50 -g110 -t30 -z11 -l -X". "-o50" sets the maximum overlap between CDS to 50 nucleotides; "-g110" sets the minimum CDS length to 110 nucleotides; "-t30" sets the threshold score for CDS to 30; "-z11" determines the NCBI translation table code "11" to specify stop codons; "-l" sets the input sequence as linear, and "-X" allows CDS to extend off the end of the input FASTA sequence.

B.2.3.2 Generation of Position Weight Matrix

A Position Weight Matrix (PWM) is generated for regions upstream of start-sites using the output from B.2.3.1. First, the script "upstream-coords.awk" is run with parameters '25 0' to extract sequence regions upstream of CDS predicted in B.2.3.1. Next, the "ELPH" program is run with parameter 'LEN=6' to create a PWM from the region upstream of the CDS start sites. ELPH is a general-purpose Gibbs sampler for finding motifs in a set of DNA or protein sequences (http://www.cbcb.umd.edu/software/ELPH/). The distribution of start codons is also generated from the output of the first iteration.

B.2.3.3 Second iteration

The second iteration of glimmer3 gene finding is run with the ICM from B.2.3.1 and the PVM from B.2.3.2 with parameters "-o50 -g110 -t30 -z11 -l –X –P <number-list>". <number-list> consists of three comma-delimited values that specify the probabilities of different start codons as determined from the output of B.2.3.2. The output includes a set of putative CDS described in a .txt summary report and coordinate files of all predicted genes in the BSML format.

C. Annotation

C.1. Translation of preliminary CDS into peptides

All predicted protein-coding genes from all scaffolds and/or contigs are translated into peptide sequences using a wrapper to the "transeq" program from the EMBOSS package [6] with the parameters "-table 11" to specify the bacterial translation table. The "-trim 1" option is also used to eliminate trailing "X" or "*" characters from the translation. A peptide FASTA file (.fsa) is provided as the output.

C.2. CDS homology searches (round 1)

Two types of homology searches are performed to generate the evidence, which is used to assign a functional annotation to each CDS: the translated CDS are compared against the UniRef100 non-redundant protein database from UniProt (http://www.uniprot.org/) using BLASTX, and against the two protein family databases Pfam and TIGRFAM [9], using HMMER [8].

C.2.1. Protein comparison and pairwise alignment

For this step, the Blast-Extend-Repraze (BER) tool (http://sourceforge.net/projects/ber/) employs a two-step process that starts with a BLASTX search followed by a modified Smith-Waterman alignment. The BER output is used to detect possible frameshifts and in-frame stop codons within the predicted CDS.

C.2.1.1. BLASTX protein comparison

The following parameters are used to perform BLASTX comparisons of all translated CDS against UniRef100: "-e 1e-5" (e-value cut-off), "-F T" (filter for low complexity regions), "-b 150" (show alignments for 150 database sequences), "-v 150" (show one-line descriptions for 150 database sequences), "-M BLOSUM62" (use BLAST matrix BLOSUM62).

C.2.1.2. Modified Smith-Waterman nucleotide sequence alignment

In order to identify frameshifts, in-frame stop codons and erroneous start codons the BER tool creates nucleotide sequence alignments of the CDS and the nucleotide sequence corresponding to the protein matches identified in C.2.1.1. For these sequence alignments the CDS are extended by 300 nucleotides upstream and downstream of the start and stop codon. Therefore, if there is a sequencing error or a natural mutation that has split one gene into two, the BER tool creates an alignment across those two fragments. BER is executed with the following parameters "-e 1e-5" (maximum e-value), -E 1e-5 (maximum p-value), "-n 150"

(maximum number of hits, similar to the BLAST -v option)), "-N 0" (maximum number of hits per region).

C.2.2. Protein family comparisons
Each translated CDS is searched against two database of Hidden Markov Models (HMMs) of protein and protein domain families, TIGRFAM (http://www.jcvi.org/cms/research/projects/tigrfams/) and Pfam (http://pfam.sanger.ac.uk/), using HMMER2 [8] with default parameters.

C.3. CDS overlap analysis
The results from the tRNA, rRNA and CDS gene calls together with the evidence generated through the HMM homology searches are used to remove overlapping genes either between CDS or between CDS and tRNA or rRNA genes. Only overlaps of at least 60 nucleotides are considered for resolution. When a CDS that has no homology evidence (HMM matches passing cutoff or BER alignments) overlaps with another CDS that does contain evidence, the CDS without evidence is removed. If both (or neither) of the CDS' show homology evidence, they are left in place. When a CDS overlaps with an RNA prediction, the RNA prediction is given a higher priority and the CDS is removed.

C.4. CDS homology searches (round 2)
After the automatic curation of start sites, the newly changed gene models are retranslated. These new polypeptides are then run through another set of BLASTX, HMM and BER searches to update similarity evidence for functional annotation. In addition, each polypeptide is run against the NCBI COG database using BLASTP with parameters "-e 1e-5 -F 'F' -b 500 -v 500 -M BLOSUM62".

C.5. Functional annotation
An in-house script is used to assign functional names, gene symbols, EC numbers and GO terms to each CDS based on a ranked hierarchy of evidence sources generated through the homology searches. The script considers sequence homology to TIGRFAM and PFAM HMMs and matches to the UniRef100 protein database. BER evidence is used to identify searches against UniRef100 with at least 35% sequence identity over 80% gene length. A series of naming rules are applied based on the type of HMM match. Equivalog HMM hits are most preferred and names are used without modification. For other HMM types protein names are appended with 'family protein' or 'domain protein' based on the isology type of the HMM. If the protein matches a hypothetical equivalog HMM the name will be 'conserved hypothetical protein'. The decision process that is used to determine functional annotations is identical to the one from the IGS Annotation Engine, which is described in detail elsewhere (K Galens et al. *submitted*).

C.5. Creation of different output files

To allow users of the pipeline to edit, visualize, and publish the annotated sequences and to submit them to the public sequence databases, the pipeline output is stored in a number of different file formats. For each annotated sequence assembly or contig .gbk and .gff files are generated, which can be opened and edited with the Artemis sequence annotation tool [13]. A .txt feature file can be loaded into the Sequin tool and used for GenBank sequence submission (http://www.ncbi.nlm.nih.gov/Sequin/index.html).

References

1. Miller JR, Koren S, Sutton G (2010) Assembly algorithms for next-generation sequencing data. Genomics 95: 315-327.

2. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 18: 821-829.

3. Delcher AL, Bratke KA, Powers EC, Salzberg SL (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. Bioinformatics 23: 673-679.

4. Lowe TM, Eddy SR (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res 25: 955-964.

5. Lagesen K, Hallin P, Rodland EA, Staerfeldt HH, Rognes T, et al. (2007) RNAmmer: consistent and rapid annotation of ribosomal RNA genes. Nucleic Acids Res 35: 3100-3108.

6. Rice P, Longden I, Bleasby A (2000) EMBOSS: the European Molecular Biology Open Software Suite. Trends Genet 16: 276-277.

7. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25: 3389-3402.

8. Eddy SR (2009) A new generation of homology search tools based on probabilistic inference. Genome Inform 23: 205-211.

9. Haft DH, Selengut JD, White O (2003) The TIGRFAMs database of protein families. Nucleic Acids Res 31: 371-373.

10. Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, et al. (2003) The COG database: an updated version includes eukaryotes. BMC Bioinformatics 4: 41.

11. Suzek BE, Huang H, McGarvey P, Mazumder R, Wu CH (2007) UniRef: comprehensive and non-redundant UniProt reference clusters. Bioinformatics 23: 1282-1288.

12. Schatz MC, Phillippy AM, Shneiderman B, Salzberg SL (2007) Hawkeye: an interactive visual analytics tool for genome assemblies. Genome Biol 8: R34.

13. Carver T, Berriman M, Tivey A, Patel C, Bohme U, et al. (2008) Artemis and ACT: viewing, annotating and comparing sequences stored in a relational database. Bioinformatics 24: 2672-2676.